

UNIVERSIDAD POLITÉCNICA SALESIANA

SEDE QUITO

CARRERA

INGENIERÍA DE SISTEMAS

Trabajo de titulación previo a la obtención del título de:

Ingenieros de sistemas

TEMA

CREACIÓN DE UNA HERRAMIENTA COLABORATIVA PARA PRÁCTICAS DE
LABORATORIO DEL CLAUSTRO DOCENTE DE PROGRAMACIÓN DE LA
UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE QUITO

AUTORES

COBO VALLE FRANCISCO JAVIER

OÑA DIAZ JEFFERSON LEONARDO

TUTOR

TUFIÑO CÁRDENAS RODRIGO EFRAÍN

Quito, julio de 2021

CESIÓN DE DERECHOS DE AUTOR

Nosotros Cobo Valle Francisco Javier con documento de identificación No. 1754383717 y Oña Diaz Jefferson Leonardo con documento de identificas. No. 1724482094, manifestamos nuestra voluntad y cedemos a la Universidad Politécnica Salesiana la titularidad sobre los derechos patrimoniales en virtud de que somos autores del trabajo de titulación intitulado: **CREACIÓN DE UNA HERRAMIENTA COLABORATIVA PARA PRÁCTICAS DE LABORATORIO DEL CLAUSTRO DOCENTE DE PROGRAMACIÓN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE QUITO**, mismo que ha sido desarrollado para optar por el título de **INGENIERO DE SISTEMAS**, en la Universidad Politécnica Salesiana, quedando la Universidad facultada para ejercer plenamente los derechos cedidos anteriormente.

En aplicación a lo determinado en la Ley de Propiedad Intelectual, en condición de autores nos reservamos los derechos morales de la obra antes citada. En concordancia, suscribo este documento en el momento que hago entrega del trabajo final en formato digital a la Biblioteca de la Universidad Politécnica Salesiana.



Cobo Valle Francisco Javier

C.I. 1754383717



Oña Diaz Jefferson Leonardo

C.I. 1724482094

Quito, julio del 2021

DECLARACIÓN DE COAUTORIA DEL TUTOR

Yo, declaro que bajo mi dirección y asesoría fue desarrollado el trabajo de titulación con el tema: CREACIÓN DE UNA HERRAMIENTA COLABORATIVA PARA PRÁCTICAS DE LABORATORIO DEL CLAUSTRO DOCENTE DE PROGRAMACIÓN DE LA UNIVERSIDAD POLITÉCNICA SALESIANA, SEDE QUITO, realizado por Francisco Javier Cobo Valle y Jefferson Leonardo Oña Diaz, obteniendo un producto que cumple con los requisitos estipulados por la Universidad Politécnica Salesiana para ser considerado como trabajo final de titulación.

A handwritten signature in blue ink, appearing to read 'R. E. Tufiño', with a long horizontal stroke extending to the right.

TUFIÑO CÁRDENAS RODRIGO EFRAÍN

C.I. 1717646390

Quito, julio del 2021

DEDICATORIA

Dedico este trabajo a mis padres Luis Cobo y Enma Valle, por su inmenso apoyo y paciencia en todos estos años, son mi motor para seguir adelante, la mejor forma de pagar su esfuerzo es plasmarlo en algo tangible y demostrar ser un profesional a la altura de sus expectativas.

Cobo Valle Francisco Javier

Dedico este trabajo a mis padres Patricio Oña y Gloria Diaz, que me han ayudado a llegar a este punto de mi carrera, ha sido un tramo muy importante para mí formación y sin ellos no lo hubiese podido lograr.

Oña Diaz Jefferson Leonardo

AGRADECIMIENTO

Agradezco a mi familia, a mi padre Luis Cobo, a mi madre Enma Valle y a mi hermano Esteban Cobo por su constante apoyo y motivación diaria.

A mi compañero y amigo Jefferson Oña por su trabajo en el presente proyecto y a nuestro tutor Tufiño Cárdenas Rodrigo Efraín por su apoyo, enseñanzas y paciencia.

Cobo Valle Francisco Javier

Agradezco a mis padres por estar siempre apoyándome con las decisiones que he tomado, a mis amigos por su infinito apoyo en toda mi carrera sobre todo a mi amigo y compañero de tesis Javier Cobo y a nuestro tutor que nos brindó un amplio conocimiento para el desarrollo del presente proyecto.

Oña Diaz Jefferson Leonardo

INDICE

INTRODUCCIÓN	1
1.1 Antecedentes.....	1
1.2 Problema de estudio	1
1.3 Justificación.....	3
1.4 Objetivos	4
1.5 Objetivos específicos.....	4
1.6 Metodología	4
CAPÍTULO II	6
MARCO TEÓRICO.....	6
2.1 Metodologías de desarrollo de software.....	6
2.1.1 Metodología SCRUM.....	6
2.1.2 Extreme Programming (XP).....	7
2.2 Lenguajes de programación	9
2.3 Marcos de trabajo	10
2.3.1 Marcos de trabajo para JavaScript.....	10
2.3.1.1 Angular.....	10
2.3.1.1 React.....	11
2.4 Base de datos	12
2.5 NodeJS	13
2.6 API.....	14
2.7 API REST.....	14

2.8 Guías de laboratorio	15
2.9 Claustro docente.....	15
CAPÍTULO III	16
ANÁLISIS Y DISEÑO	16
3.1 Análisis de viabilidad	16
3.1.1 Viabilidad técnica.....	16
3.1.2 Viabilidad económica.....	17
3.1.3 Viabilidad operacional.....	18
3.2 Análisis de requerimientos	18
3.2.1 Alcance	19
3.2.2 Definiciones, acrónimos y abreviaturas.....	21
3.2.3 Descripción general	21
3.2.3.1. Perspectiva del producto	21
3.2.3.2. Funciones del producto	22
3.2.3.3. Características de usuarios	22
3.2.3.4. Restricciones.	22
3.2.3.5. Suposiciones y dependencias.	23
3.2.4 Requerimientos específicos	23
3.2.5 Requisitos funcionales.....	24
3.2.5.1. Requisito Funcional (RF1).....	24
3.2.5.2. Requisito Funcional (RF2).....	25
3.2.5.3. Requisito Funcional (RF3).....	25

3.2.5.4. Requisito Funcional (RF4).....	26
3.2.5.5. Requisito Funcional (RF5).....	27
3.2.5.6. Requisito Funcional (RF6).....	27
3.2.5.7. Requisito Funcional (RF7).....	28
3.2.5.8. Requisito Funcional (RF8).....	28
3.2.5.9. Requisito Funcional (RF9).....	29
3.2.5.9. Requisito Funcional (RF11).....	30
3.2.6 <i>Requisitos no funcionales</i>	31
3.2.6.1. Requisitos de rendimiento.....	31
3.2.6.2. Seguridad.....	31
3.2.6.3. Disponibilidad.....	32
3.2.6.4. Escalabilidad.....	32
3.3 Diseño.....	32
3.3.1 <i>Diagramas de casos de uso</i>	32
3.3.2 <i>Diagramas de secuencia</i>	36
3.3.3. <i>Diagrama conceptual de base de datos</i>	41
3.3.4 <i>Diagrama de clases</i>	44
3.3.5 <i>Mapas de navegación</i>	45
3.3.6 <i>Interfaces abstractas</i>	47
3.3.6.1. Interfaces abstractas comunes para todos los usuarios.....	48
2.3.6.3. Interfaces abstractas del Coordinador de asignatura.....	51
CAPÍTULO III.....	57

CONSTRUCCIÓN Y PRUEBAS	57
4.1 Herramientas	57
4.1.1 <i>Git</i> Hub.....	57
4.1.2 <i>Visual Studio Code</i>	57
4.1.3 <i>PostMan</i>	57
4.1.4 <i>MongoDB Compass</i>	58
4.2 Product Backlog	58
4.3 Sprints.....	60
4.4 Sprint 1	60
4.4.1 <i>Preparación del entorno</i>	61
4.4.1.1. Entorno de MongoDB.	61
4.4.1.2. Entorno de NodeJS.....	61
4.5 Sprint 2	61
4.6 Sprint 3	63
4.6.1 <i>Conexión a la base en MongoDB Atlas</i>	64
4.6.2 <i>Encriptación de contraseña</i>	64
4.6.3 <i>Verificación de seguridad mediante token</i>	65
4.6.4 <i>Entregables de sprint 3</i>	65
4.7 Sprint 4	67
4.7.1 <i>Comunicación entre frontend y backend</i>	67
4.7.2 <i>Comunicación entre frontend y backend</i>	68
4.8 Sprint 5	71

4.8.1 Operación insertar	71
4.8.2 Operación buscar.....	72
4.8.3 Operación actualizar.....	73
4.8.4 Operación eliminar	73
4.9 Sprint 6	76
4.9.1 Validar permisos.....	76
4.10 Sprint 7	79
4.10.1 Operación patch.....	79
4.11 Sprint 8	82
4.11.1 Creador de ejercicios.....	82
4.11.2 Subir imágenes al ejercicio	83
4.12 Sprint 9	86
4.12.1 Creación guía del laboratorio.....	87
4.13 Sprint 10	89
4.13.1 Creación de guía de laboratorio en pdf.....	89
4.14 Sprint 11	91
4.14.1 Generación de gráficas para reporte	91
4.14.2 Generación de gráficas para reporte	92
4.15 Pruebas	92
4.15.1 Pruebas de caja negra.....	92
4.15.2 Pruebas del código.....	96
4.15.3 Pruebas de estrés	98

4.16 Análisis de resultados	100
4.17 Diagrama de implementación.....	101
CAPITULO V	102
IMPLEMENTACIÓN	102
5.1. Build de la aplicación frontend	102
5.2. Despliegue de backend.....	102
5.3. Despliegue en internet.....	103
5.4 Aplicativo funcionando en internet	105
CONCLUSIONES	106
RECOMENDACIONES	108
REFERENCIAS	109
ANEXOS.....	113

INDICE DE TABLAS

Tabla 1 Comparación entre metodologías Scrum y XP.....	8
Tabla 2 Comparación lenguajes de programación.....	9
Tabla 3 Detalles de hardware disponibles	16
Tabla 4 Detalles de software disponibles	17
Tabla 5 Costo de recursos.....	17
Tabla 6 Persona involucrada.....	19
Tabla 7 Persona involucrada.....	20
Tabla 8 Persona involucrada.....	20
Tabla 9 Requerimiento específico 1	24
Tabla 10 Requerimiento específico 2	25
Tabla 11 Requerimiento específico 3	25
Tabla 12 Requerimiento específico 4	26
Tabla 13 Requerimiento específico 5	27
Tabla 14 Requerimiento específico 6	27
Tabla 15 Requerimiento específico 7	28
Tabla 16 Requerimiento específico 8	28
Tabla 17 Requerimiento específico 9	29
Tabla 18 Requerimiento específico 10	30
Tabla 19 Requerimiento específico 11	30

Tabla 20 <i>Product Backlog</i>	58
Tabla 21 Tareas de sprint 1	60
Tabla 22 Tareas de sprint 2	62
Tabla 23 Tareas de sprint 3	63
Tabla 24 Tareas de sprint 4	67
Tabla 25 Tareas de sprint 5	71
Tabla 26 Tareas de sprint 6	76
Tabla 27 Tareas de sprint 7	79
Tabla 28 Tareas de sprint 8	82
Tabla 29 Tareas de sprint 9	86
Tabla 30 Tareas de sprint 10	89
Tabla 31 Tareas de sprint 11	91
Tabla 32 Prueba de caja negra 1	93
Tabla 33 Prueba de caja negra 2	93
Tabla 34 Prueba de caja negra 3	94
Tabla 35 Prueba de caja negra 4	95

INDICE DE FIGURAS

Figura 1 Numero de descargas NPM Trends	12
Figura 2 Arquitectura RESTAPI	14
Figura 3 Caso de uso para usuario Administrador	33
Figura 4 Caso de uso para usuario Coordinador de asignatura.....	34
Figura 5 Caso de uso para usuario Docente	35
Figura 6 Diagrama de secuencia: Crear usuario	36
Figura 7 Diagrama de secuencia: Crear Ejercicio.....	37
Figura 8 Diagrama de secuencia: Crear guía de laboratorio	38
Figura 9 Diagrama de secuencia: Generar guía de laboratorio	39
Figura 10 Diagrama <i>de secuencia: Editar perfil</i>	40
Figura 11 Diagrama de secuencia: Generar reporte	41
Figura 12 Diagrama conceptual de base de datos	43
Figura 13 Diagrama de clases de todo el aplicativo.....	44
Figura 14 <i>Mapa de navegación: Administrador</i>	45
Figura 15 <i>Mapa de navegación: Coordinador de asignatura</i>	46
Figura 16 Mapa de navegación: Docente.....	47
Figura 17 Interfaz abstracta: inicio de sesión.....	48
Figura 18 Interfaz abstracta: perfil de usuario	48
Figura 19 <i>Interfaz abstracta: gestionar carreras</i>	49

Figura 20 Interfaz abstracta: gestionar periodos	49
Figura 21 <i>Interfaz abstracta: gestionar asignaturas</i>	50
Figura 22 Interfaz abstracta coordinador de asignatura: Agregar temas	51
Figura 23 <i>Interfaz abstracta: creación de docente y asignación de coordinador</i>	51
Figura 24 <i>Interfaz abstracta coordinador de asignatura: Mostrar ejercicio</i>	52
Figura 25 <i>Interfaz abstracta coordinador de asignatura: Crear ejercicios</i>	53
Figura 26 <i>Interfaz abstracta coordinador de asignatura: Ver práctica</i>	54
Figura 27 <i>Interfaz abstracta coordinador de asignatura: Crear ejercicios</i>	55
Figura 28 <i>Interfaz abstracta coordinador de asignatura: Generar práctica</i>	56
Figura 29 Código conexión a base de datos.....	64
Figura 30 Código encriptado de clave usuario.....	64
Figura 31 Código de inicio de sesión mediante token	65
Figura 32 Pantalla de login	65
Figura 33 Registro de usuario	66
Figura 34 Comunicación entre interfaces	68
Figura 35 Documentación API.....	68
Figura 36 Estructura documento APIs	69
Figura 37 Documentación de APIs	70
Figura 38 Operación Insertar	72
Figura 39 Operación buscar	72

Figura 40 Operación actualizar	73
Figura 41 Operación eliminar	74
Figura 42 Creación de periodo	74
Figura 43 Pantalla de activación de periodo	75
Figura 44 Pantalla de creación de carreras.....	75
Figura 45 Validar usuario administrador	77
Figura 46 Creación de asignatura.....	77
Figura 47 Pantalla creación asignatura	78
Figura 48 Asignar docentes	80
Figura 49 Pantalla creación asignatura	80
Figura 50 Pantalla de bibliografías	81
Figura 51 Pantalla creación referencias	81
Figura 52 Editor tinymce	83
Figura 53 Carga de imágenes	84
Figura 54 Formato de ejercicios.....	84
Figura 55 Pantalla de creación de ejercicios	85
Figura 56 Pantalla de listado de ejercicios.....	85
Figura 57 Creación guía de laboratorio.....	87
Figura 58 Pantalla de creación de guías de laboratorio.....	88
Figura 59 Generación de guías de laboratorio	90

Figura 60	Pantalla de selección de ejercicios.....	90
Figura 61	Generación de gráficas	91
Figura 62	Pantalla de reportes.....	92
Figura 63	Resumen de las pruebas de código fuente	96
Figura 64	Duplicidad de líneas de código.....	97
Figura 65	Resultado de prueba de seguridad	97
Figura 66	Código no usado	98
Figura 67	Reporte resumen proceso crear ejercicios	99
Figura 68	Reporte resumen proceso crear guía de laboratorio	99
Figura 69	Reporte resumen proceso crear ejercicios	100
Figura 70	Reporte resumen proceso crear guía de laboratorio	100
Figura 71	Diagrama de implementación	101
Figura 72	Variables de entorno frontend	102
Figura 73	Variables de entorno backend	103
Figura 74	Código configuración frontend.....	104
Figura 75	Código configuración backend	104
Figura 76	Página de inicio de sesión del aplicativo	105

RESUMEN

El presente trabajo de titulación engloba las fases de análisis, diseño y construcción de un aplicativo web para la gestión de guías de laboratorio del claustro docente de programación de la universidad politécnica salesiana, sede Quito campus sur.

El objetivo es brindar a los docentes del claustro de programación un ambiente estandarizado de fácil acceso para la creación y gestión de sus guías de laboratorio en cada periodo académico, disminuyendo así el tiempo de creación y aumentando la utilidad de sus guías a los estudiantes.

El sistema fue desarrollado con tecnologías que se acoplen a la naturaleza web del aplicativo, tales como JavaScript, React, entre otras, las cuales se seleccionaron mediante un análisis de ventajas y desventajas, detalladas en el documento.

El aplicativo ha sido desarrollado de acuerdo con los requerimientos funcionales y no funcionales establecidos en reuniones con los interesados, para brindar un producto final de fácil acceso, intuitivo, rápido y adaptable, gracias a su naturaleza web y tecnología usada.

ABSTRACT

This degree work encompasses the phases of analysis, design and construction of a web application for the management of laboratory guides for the programming teaching staff of the Salesian Polytechnic University, Quito south campus.

The objective is to provide teachers of the programming faculty with a standardized environment of easy access for the creation and management of their laboratory guides in each academic period, reducing the creation time and increasing the usefulness of their guides to students. students.

The system was developed with technologies that are coupled to the web nature of the application, such as JavaScript, React, among others, which were selected through an analysis of advantages and disadvantages, detailed in the document.

The application has been developed in accordance with the functional and non-functional requirements established in meetings with the interested parties, to provide a final product that is easy to access, intuitive, fast and adaptable, thanks to its web nature and used technology.

INTRODUCCIÓN

Este capítulo contiene una descripción del problema de estudio cimentado en la revisión de proyectos similares, con el objetivo de tener una idea clara del problema y justificar la realización del proyecto. Aunado a ello, se encuentran los objetivos.

1.1 Antecedentes

Los docentes pertenecientes al claustro docente de programación de la Universidad Politécnica Salesiana en cada periodo académico realizan guías de laboratorio las cuales fungen como herramientas de aprendizaje para los estudiantes, dichas guías de laboratorio son creadas por cada docente al inicio del periodo y enviadas al coordinador de la asignatura para su recolección y almacenamiento, este proceso se realiza de manera manual, mediante OneDrive. A pesar de que cada guía no cambia en gran medida entre periodos académicos su creación se torna tediosa y lleva más tiempo del necesario, se pretende que este proceso sea más fácil de realizar y de tener constancia de los ejercicios que se usan entre periodos para evitar repitencia de los mismos y reforzar el conocimiento de los conocimientos impartidos a los estudiantes.

El uso de aplicaciones se ha convertido en algo cotidiano tanto en actividades lúdicas como educativas, es así como su uso en la educación es enriquecedor para la enseñanza, en especial el uso de aplicaciones de fácil acceso por medio de la web.

1.2 Problema de estudio

En el ámbito académico el uso de herramientas colaborativas basadas en tecnología es algo común en la actualidad, su uso ha sido una estrategia importante en clases (Coyle, 2007), prueba de ello son las herramientas de mensajería como el correo electrónico. “La tecnología más común y de fácil comprensión para la colaboración es el correo electrónico” (Duarte y

Tennant-Snyder, 2000). Estas habilitan funciones para facilitar el intercambio de información entre los miembros colaboradores. Dentro de cualquier organización la gestión del flujo de mensajes representa un aspecto importante a considerar, sobre todo si la colaboración implica el trabajo sobre un mismo material. Dando como resultado la necesidad de un gestor de control de versiones sobre los archivos, y este último punto es un verdadero desafío si se lo trata de solventar por medios poco flexibles como el correo electrónico o mensajes instantáneos (Massey, 2008). Sin embargo, el uso de herramientas colaborativas en la nube se ha convertido en una solución común, tanto para guardar información como para colaborar sobre archivos comunes. Ejemplificando lo anterior, aplicativos de Google (Google Drive) y Microsoft (OneDrive), representan un 74% de las tecnologías usadas en dichos escenarios, 62% y 12% respectivamente (Agus, Destiawati y dhika, 2019). Lo que ha extendido el uso de dichas herramientas como una alternativa viable para solventar problemas generales (Kai-Wai Chu y Kennedy, 2011). Sin embargo, no ofrece soluciones enfocadas en un grupo académico con interés en común. Lo que genera aplicativos generales estáticos que solo fungen como meros bancos de datos para guardar información. Los cuales no aportan valor a procesos auxiliares en el aprendizaje de los alumnos.

En Hong Kong en el año 2010, se describió el uso de herramientas colaborativas como Google Docs para la colaboración entre participantes educativos, con la participación de 22 estudiantes de Pregrado en la Universidad de Hong Kong. Por medio de cuestionarios, se generan los siguientes resultados. La mayoría de los usuarios tuvieron experiencias positivas 3.5 de satisfacción sobre 5, considerado como buena. Concluyendo así que el uso de herramientas colaborativas otorga una gestión efectiva de activos por parte del docente (Kai-Wai Chu y Kennedy 2011).

En España en el año 2017, se implementaron herramientas colaborativas en red en este caso un blog para los estudiantes de informática, los cuales al iniciar el periodo lectivo se registraban para enviar las prácticas, para reducir la carga por parte de los docentes. Dando como resultado una experiencia positiva a comparación de años anteriores, esto debido a que anteriormente los profesores reciben todas las prácticas diarias al correo por parte de sus alumnos, a lo cual el docente procede a descargar dichas prácticas tomando en cuenta que los alumnos podían enviarla a cualquier hora del día y los correos podían mezclarse con temas no relacionados (Otero, J., Sanz y Otero, C., 2017).

Claramente existen herramientas que ayudan a la gestión de recursos académicos y solventan en gran medida las necesidades de un público general, pero en casos puntuales esto no es así, siendo preciso la creación de aplicativos específicos para satisfacer dichas necesidades. Es por eso por lo que el desarrollo de una aplicación web para la gestión de Prácticas de Laboratorio dirigido al Claustro Docente de Programación (CDP) de la Universidad Politécnica Salesiana (UPS), resulta relevante para dar solución a dos problemas que tiene el CDP actualmente: la estandarización del formato en las Prácticas de Laboratorio y garantizar no repetir los problemas de programación usadas en evaluaciones durante un mismo período académico.

1.3 Justificación

El presente proyecto busca proponer una solución a herramientas de colaboración tradicionales, con ello, se plantea la gestión y creación de documentos, mediante el concepto de herramienta colaborativa, minimizando así el tiempo de creación de documentos y facilitando la gestión de los mismos. Frente a la problemática de tener documentos no estandarizados y distribuidas en varios archivos por cada docente del Claustro Docente de

Programación, pues la gran mayoría de herramientas actuales, no se adaptan a las necesidades específicas de un grupo con intereses en común.

El desarrollo del aplicativo web tiene como propósito la creación de una herramienta colaborativa web rápida e intuitiva, la cual permita a los docentes del Claustro Docente de Programación, la creación y gestión de sus guías de laboratorio. Las cuales podrán ser descargadas en un formato preestablecido con una estructura estandarizada.

1.4 Objetivos

Crear una herramienta colaborativa en internet que permita gestionar las guías de laboratorio del Claustro Docente de Programación de la Universidad Politécnica Salesiana.

1.5 Objetivos específicos

- Analizar los requerimientos del Claustro Docente de Programación con respecto al manejo de las prácticas de laboratorio.
- Diseñar el aplicativo web en base a los requerimientos de los miembros del Claustro Docente de Programación.
- Construir el aplicativo web con el uso de la metodología ágil SCRUM, basado en tecnologías de desarrollo web.
- Implementar el aplicativo en un servidor en Internet.
- Evaluar el aplicativo web mediante el uso de métricas.

1.6 Metodología

El proyecto se desarrolló mediante la metodología SCRUM, con el soporte de herramientas de investigación como la entrevista para obtener los requerimientos, se decantó el uso de una metodología ágil debido a que el desarrollo del proyecto debía ser realizado en un

corto periodo de tiempo y se preveía que los requerimientos del Product Owner (Coordinador del claustro docente) iban a ser cambiantes. El uso de SCRUM brindó flexibilidad y seguimiento en la ejecución de las tareas, gracias a las reuniones continuas con el cliente se garantizó que el aplicativo cumpla con las necesidades del cliente semana tras semana.

Definición de roles para SCRUM

En base a la metodología, a continuación, se muestra el personal asignado a cada rol dentro del desarrollo.

- Product Owner: este rol estuvo a cargo del coordinador del claustro docente de programación Tufiño Cárdenas Rodrigo Efraín, el cual conoce el flujo de trabajo de las guías de laboratorio, quien fijó los requerimientos del aplicativo.
- Scrum Master: A su vez Tufiño Cárdenas Rodrigo Efraín, tutor del proyecto fue el encargado de organizar el equipo.
- Equipo de desarrollo: asignado a Cobo Javier y Oña Jefferson, encargados de plasmar los requerimientos del usuario en el aplicativo.
- Usuarios: se asignaron tres perfiles, los que van a interactuar con la aplicación, siendo estos: administrador, docente y estudiante.

Instrumentos y herramientas

El levantamiento de requerimientos se realizó por medio de la técnica de investigación entrevista, los requerimientos fueron plasmados en historias de usuario, con las cuales se construyó el Product Backlog del cual se derivarán los Sprint Backlog de cada iteración de trabajo. El seguimiento de cada iteración se realizó mediante la herramienta web Jira para mantener orden y constancia del trabajo realizado.

CAPÍTULO II

MARCO TEÓRICO

Este capítulo contiene una descripción de metodologías, lenguajes de programación, herramientas y bases de datos para el desarrollo de software orientado a la web, con el objetivo de seleccionar la opción más adecuada en base al análisis de literatura relacionada.

2.1 Metodologías de desarrollo de software

Metodología de desarrollo de software es un conjunto de métodos y prácticas coherentes y relacionadas entre sí, las cuales dan incremento al desarrollo de software. (Rivas et al, 2015).

El seguir una metodología permite que el desarrollo siga pautas ya establecidas, asegurando así una estructura en el trabajo, habilitando a los desarrolladores una guía para organizar el trabajo a realizar, garantizando así la obtención de un producto final que sea capaz de satisfacer las necesidades del cliente final.

Según Rivas et al. (2015) actualmente el desarrollo de software está sujeto a un entorno rápidamente cambiante, por ello se debe dar soluciones rápidas y ágiles con el objetivo de responder con un producto de calidad y que satisfaga las necesidades del cliente. Estas metodologías son:

- Programación extrema (XP).
- SCRUM.

2.1.1 Metodología SCRUM

SCRUM es uno de los marcos de metodologías ágiles para desarrollo de productos los cuales pueden admitir cambios que surgen en cualquier fase del proceso del desarrollo, estos

pueden surgir en los requerimientos o estar basados en el mercado. Adaptándose a las necesidades cambiantes o no totalmente claras del usuario final y del mercado.

A diferencia de los equipos tradicionales donde todos trabajan para hacer el producto final y luego entregarlo. En las metodologías ágiles existen pequeños grupos de trabajo, los cuales tendrán un tiempo para desarrollar un pequeño requerimiento funcional de la lista completa de requerimientos, este tiempo de desarrollo corto se lo conoce como iteración.

Esto se hace con el fin de llevar un mejor control de lo que se desarrolla y quien lo desarrolla. Además, el tiempo de duración de cada iteración se desarrolla con un objetivo claro, presentado al cliente para recibir retroalimentación y con ello generar mejoras continuas.

SCRUM se enfoca en resolver cualquier tipo de problema, sea grande o pequeño de forma iterativa y no necesariamente tiene que ser un problema de software, esto se lleva a cabo uniendo al equipo para realizar un objetivo primordial mientras colaboran entre ellos.

Los pasos que sigue la metodología SCRUM son los siguientes. i) Definir Equipo SCRUM, ii) Definir duración de Sprint, iii) Nombrar SCRUM Master, iv) Identificar Product Owner, v) Crear Backlog, vi) Planificar y ejecutar Sprint, vii) Cerrar Sprint actual, viii) Planificar siguiente Sprint.

2.1.2 Extreme Programming (XP)

Wells define la define como una metodología de desarrollo de software ágil basado en la simplicidad, colaboración y reutilización del código con el fin de producir software de calidad el cual satisfaga las necesidades del usuario final, adicionalmente proporciona un ambiente de desarrollado adecuado para el equipo de trabajo. (Wells, 2001).

Dicha metodología se enfoca en la prueba y error para el desarrollo, permitiendo la participación del cliente en todo el proceso, promoviendo el trabajo en parejas para el desarrollo

con el objetivo de fortalecer el equipo y promover un ambiente laboral agradable (Salazar et. al, 2015).

Los principios de XP deben ser plasmados en prácticas para el desarrollo de la metodología. Rosado et. al (2012) indica que los principios son plasmados en cuatro prácticas fundamentales.

- Entregas limitadas
- Cliente en el sitio
- Programación en pareja
- Semana de trabajo

Con el fin de determinar la metodología a usar se planteó una tabla comparativa entre la metodología SCRUM y XP.

Tabla 1

Comparación entre metodologías Scrum y XP

Descripción	Metodologías	
	SCRUM	XP
Estilo del trabajo	Los miembros del equipo trabajan de forma individual.	La programación se realiza en parejas.
Tiempo máximo entre iteraciones	Cada iteración puede durar entre una a cuatro semanas.	Cada iteración puede durar entre una a tres semanas.
Cambios a tareas ya completadas	Funcionalidades son entregadas y aceptadas por el cliente	Funcionalidades ya entregadas son susceptibles al cambio.
Prioridad en las tareas	Prioridades pueden cambiar de acuerdo con el criterio de los involucrados.	Las prioridades las define el cliente.
Metodología	Metodología ágil basada en la administración del proyecto.	Metodología ágil basada más enfocada en la programación.

Nota. Tabla comparativa entre metodologías Scrum y XP. Elaborado por: los autores.

La metodología SCRUM tiene la ventaja de que las tareas pueden priorizarse en base al criterio del equipo, brindando flexibilidad al desarrollo. La metodología XP es más sensible a cambios en entregables ya realizados los cuales previamente fueron aprobados por el cliente,

generando así tiempo extra en el desarrollo, lo cual no se busca en el presente proyecto, ya que el fin es adaptarse a las necesidades del usuario en el menor tiempo de desarrollo. Las características presentadas en la Tabla 1 y el análisis realizado nos lleva a la selección de la metodología SCRUM para el presente proyecto.

2.2 Lenguajes de programación

2.2.1 JavaScript

JavaScript está diseñada para poder ser incrustada en código HTML, es un lenguaje interpretado por el navegador como código fuente, por ende, no es necesario compilar los programas para ejecutarlos (Maza, 2012).

Según Flanagan (2007), JavaScript es el lenguaje más utilizado en la elaboración de páginas web con una sintaxis similar a Java.

Similar a PHP con una diferencia notable, la cual consiste en que los procesos JavaScript son ejecutados del lado del cliente en el navegador web, por ende, no hay un intercambio de datos donde se involucre el servidor (Mohedano et. al, 2012).

En la Tabla 2 a pesar de que todos los lenguajes de programación poseen características similares, se decantó por JavaScript por ser muy adecuado y compatible con el marco de trabajo y base de datos a usar en el proyecto.

Tabla 2

Comparación lenguajes de programación

Descripción	Lenguajes de programación		
	JavaScript	Python	PHP
Nivel de abstracción	Alto nivel	Alto nivel	Alto nivel
Multiplataforma	Si	Si	Si
Herencia	Si	Si	Si

Polimorfismo	Si	Si	Si
Orientado al desarrollo de aplicaciones web	Si	Si	Si
Compilado o interpretado	Interpretado	Interpretado	Interpretado
Expansión de funcionalidades con librerías propias	Si	Si	Si

Nota. Comparación entre lenguaje de programación. Elaborado por: los autores.

Las características de los lenguajes de programación descritos en la Tabla 2, presentan similitud de características, lo cual lleva a pensar que el uso de cualquier lenguaje es adecuado, pero debido a que el ambiente de desarrollo es altamente compatible con JavaScript y que los desarrolladores están familiarizados y poseen conocimientos en su uso, se optó por seleccionar dicho lenguaje, debido a que no se invertirá tiempo en aprender y posiblemente se omitirán errores, gracias a experiencias previas en su uso.

2.3 Marcos de trabajo

Un marco de trabajo es un conjunto de prácticas, reglas y criterios, esto ofrece al usuario buenas prácticas en el código del aplicativo, adicionalmente los marcos de trabajo proporcionan al desarrollador un conjunto de funciones ya creadas comunes entre aplicaciones como por ejemplo el inicio de sesión (Sucre, 2014).

Lo mencionado anteriormente no solo reduce el tiempo de trabajo del equipo de desarrollo sino también ayuda a facilitar el trabajo en equipo, ya que si más de una persona trabaja en el mismo aplicativo las reglas dadas por el marco de trabajo hará que todo el equipo desarrolle con una lógica en común.

2.3.1 Marcos de trabajo para JavaScript

2.3.1.1 Angular. Angular es un marco de trabajo escrito en HTML y TypeScript, este último es un lenguaje de programación de código abierto el cual permite la opción de escritura

estática en JavaScript. Las aplicaciones desarrolladas con Angular están compuestas por los denominados módulos Ng, los cuales son contenedores de bloques de código (Saks, 2019).

2.3.1.1 React. Utiliza una sintaxis JSX, la cual es una mezcla entre JavaScript y HTML, similar a Angular, con la excepción de que en React, JavaScript tienen mayor relevancia y se aprovecha mejor sus capacidades.

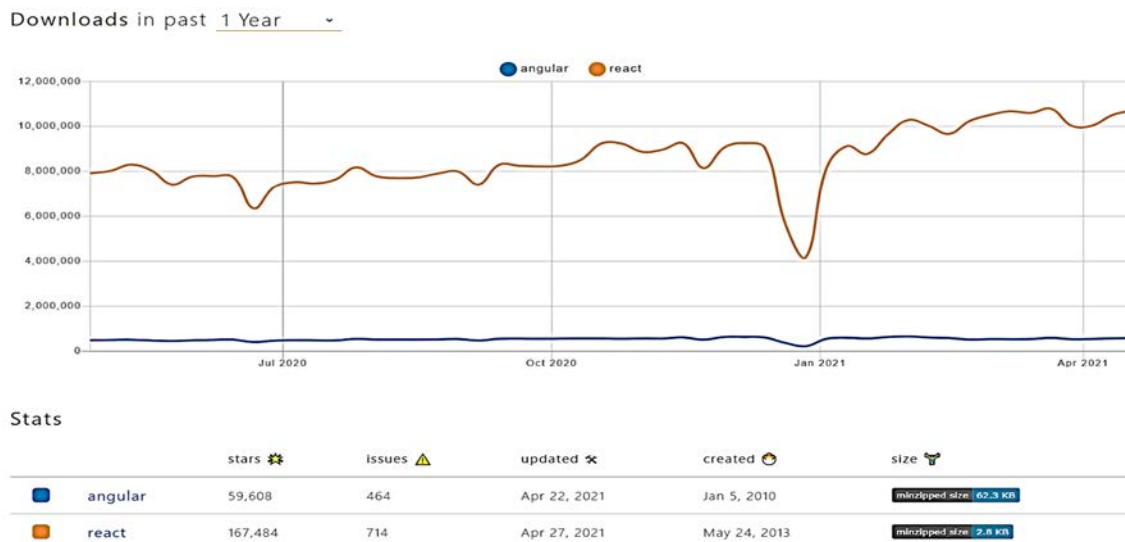
JSX se utiliza para construir elementos los cuales forman los denominados componentes de React, estos elementos son objetos simples y fáciles de crear, estos elementos son actualizados en el virtual DOM (Modelo de representación de objetos en páginas web), actualizando únicamente los elementos que han sido actualizados por el cliente.

Los componentes de React son los componentes básicos de las aplicaciones de React, permiten que la interfaz de usuario se divida en partes que son independientes entre sí y pueden ser reutilizadas. Los datos se pueden pasar entre componentes, cada componente puede ser renderizado en el aplicativo cada vez que se actualiza (Saks, 2019).

Con el fin de obtener un punto de referencia adicional en el marco de trabajo a utilizar se realizó una búsqueda de las tendencias actuales en NPM Trends, para identificar que tan popular es entre los usuarios, dando como resultado una clara preferencia por React, con un aproximado de 10 millones de descargas entre abril de 2020 y abril de 2021 Ver Figura 1.

Figura 1

Numero de descargas NPM Trends



Nota. Número de descargas y popularidad entre React y Angular. Fuente: NPM Trends

El análisis de tendencias en el uso de un marco de trabajo por parte de los usuarios ofrece un punto de partida importante debido a que mientras más usuarios lo usen significa que habrá más información y código ya probado para el uso por parte de los desarrolladores, permitiendo tiempos menores de trabajo y menos susceptibilidad a cometer errores. Adicionalmente el equipo de trabajo posee conocimientos en React permitiendo que no se use tiempo en aprendizaje. Sin mencionar que el ambiente es altamente compatible.

2.4 Base de datos

2.4.1 Base de datos no relacional

Una base de datos no relacional no está atada al modelo tradicional de base de datos relacional, no es estricta con su estructura de manejo de datos por tablas y relaciones, esta se caracteriza por no tener un esquema definido, son frecuentemente usadas por aplicaciones web, donde se garantiza la disponibilidad. (Castillo, et. al, 2017).

2.4.2 MongoDB

MongoDB siendo una base de datos no relacional no usa tablas en su estructura de datos en contraparte se usan documentos, dichos documentos son de tipo JSON (JavaScript Object Notatio). MongoDB almacena los documentos en un tipo llamado BSON (JSON Binario), permitiendo una integración fácil y rápida con ciertas aplicaciones (Castillo, et. al, 2017). Los documentos son gestionados por colecciones, donde tienen el siguiente esquema.

- Nombre de campo
- Tipo de datos
- Valor

La elección de MongoDB como base de datos para el proyecto se debe a que todo el entorno de desarrollo es compatible con JavaScript tanto el lenguaje de programación como el marco de trabajo, adicionalmente el uso de una base de datos no relacional permite gran escalabilidad al aplicativo, brindando flexibilidad y facilidad en el desarrollo.

2.5 NodeJS

Posteriormente se presentó el análisis y elección de JavaScript como lenguaje de programación para el desarrollo del aplicativo, por ende, la elección más adecuada para un lenguaje de programación de servidor es NodeJS, debido a que es un entorno de ejecución de JavaScript.

Lo más notable de NodeJS es que todas peticiones son atendidas por un único hilo de ejecución, el cual no se detiene a pesar de que existan peticiones las cuales requieren un resultado de otro proceso, esto garantiza la ejecución continua de peticiones y se denomina código no bloqueante o asíncrono (Rodríguez, 2017).

Adicionalmente posee un repositorio de módulos NPM extenso y de fácil acceso, el cual permite añadir funcionalidades que ahorren tiempo y garanticen compatibilidad.

2.6 API

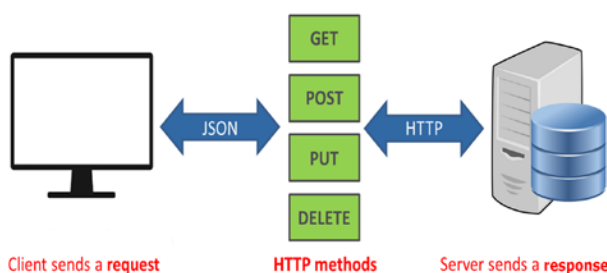
La abreviatura API tiene por significado Interfaz de Programación, la cual se puede definir como un conjunto de protocolos, la cual tienen como objetivo comunicar dos aplicaciones entre sí por medio de reglas, dicha comunicación tiene como fin dar acceso a funciones que cumplen acciones específicas.

2.7 API REST

REST es una arquitectura para la implementación de aplicaciones web, se caracterizan por manipular recursos mediante métodos HTML, estos son: GET, POST, PUT, DELETE para realizar acciones de leer, crear, editar y eliminar respectivamente, por lo general las respuestas de una API REST son en formato JSON.

Figura 2

Arquitectura RESTAPI



Nota. Arquitectura Restfull entre un cliente y un servidor. Fuente: <https://aprendiendoarduino.wordpress.com>.

A continuación, se menciona las bases teóricas más relevantes sobre los elementos relacionados al área donde será usado el aplicativo.

2.8 Guías de laboratorio

Las guías de laboratorio son documentos usados por los docentes que fungen como una guía para realizar tareas prácticas con el fin de evaluar los conocimientos teóricos impartidos por el docente en sus clases, estas guías contienen ejercicios o problemas prácticos que serán resueltos por los estudiantes con los conocimientos adquiridos.

2.9 Claustro docente

Los claustros docentes son entidades formados por docentes de la Universidad Politécnica Salesiana, dichos docentes comparten intereses académicos, donde el trabajo colaborativo es una herramienta de crecimiento personal y profesional. El claustro es de libre acceso para cualquier docente que quiera participar.

CAPÍTULO III

ANÁLISIS Y DISEÑO

Este capítulo contiene el análisis de viabilidad del proyecto, se describen los requerimientos funcionales del aplicativo web, adicionalmente se presenta el diseño del aplicativo mediante diagramas UML.

3.1 Análisis de viabilidad

Para el presente análisis se han establecido tres aspectos dentro del desarrollo de software, siendo estos: viabilidad técnica, económica y operacional.

3.1.1 Viabilidad técnica

El aplicativo está desarrollado con herramientas de código abierto, la plataforma sobre la que funcionará no es relevante, debido a que el aplicativo es web, siendo independiente del Sistema Operativo del usuario, solo requiriendo el uso de un navegador web. El hardware y software involucrados en el desarrollo tienen las siguientes características.

En la Tabla 3 se puede observar el nombre del dispositivo, la cantidad, las funciones que cumplirá dentro del desarrollo y la descripción técnica de los componentes de hardware.

Tabla 3

Detalles de hardware disponibles

Requisitos de hardware			
Dispositivo	Cantidad	Funciones	Descripción
Ordenador portátil HP	1	Desarrollo, pruebas, base de datos	<ul style="list-style-type: none">• Procesador: Intel Core I7• Memoria: 8GB RAM• Disco: 256GB Disco Duro solido• Arquitectura: 64bits
Ordenador de sobremesa Dell	1	Desarrollo, pruebas, base de datos	<ul style="list-style-type: none">• Procesador: Intel Core I4• Memoria: 8Gb RAM• Disco: 1 TB• Arquitectura: 64bits

Nota. Requisitos de hardware para el desarrollo. Elaborado por: los autores.

En la Tabla 4 se puede observar el nombre del producto, la cantidad, las funciones que cumplirá dentro del desarrollo y el tipo de licencia.

Tabla 4

Detalles de software disponibles

Requisitos de software			
Producto	Cantidad	Funciones	Descripción
Amazon web services	1	Alojamiento de aplicación web	Recursos del grupo de investigación
Windows	2	Sistema operativo	Licencia: Windows 10 Pro
Visual Studio Code	2	Editor de código	Licencia: MIT
GitHub	2	Repositorio de código	Licencia: Free
MongoDB	1	Sistema de base de datos	Licencia: Free
Opera	2	Navegador Web	Licencia: Free

Nota. Requisitos de software para el desarrollo del aplicativo web. Elaborado por: los autores.

3.1.2 Viabilidad económica

La inversión para el desarrollo del aplicativo está directamente relacionada al pago del salario de un ingeniero de desarrollo. Se ha fijado un total de \$800 por cada 8 horas de trabajo, con un tiempo de desarrollo aproximado de seis meses.

Para dichos costos se han establecido diferentes ámbitos económicos, los cuales abarcan costo de equipos, licenciamiento, comunicación, costo de personal y gastos varios en general. Dichos costos se pueden ver de manera global en la Tabla 5, para un mayor detalle de los costos se puede visualizar en los anexos (Viabilidad económica).

Tabla 5

Costo de recursos

Referencia	Valor
Recursos de hardware	\$1.904,00
Recursos de software	\$179,20
Recursos de comunicación	\$403,20
Recursos humanos	\$11894,40
Recursos extra relacionados	\$1100,00
Total	\$15.480,00

Nota. Detalles de costos para el desarrollo del aplicativo web. Elaborado por: los autores.

Todos los costos del desarrollo de la aplicación serán cubiertos por los estudiantes autores del proyecto.

3.1.3 Viabilidad operacional

Actualmente los miembros pertenecientes al Claustro Docente de Programación hacen uso de la herramienta colaborativa Microsoft SharePoint, con el fin de compartir las prácticas de laboratorio. El coordinador del claustro es el encargado de gestionar las guías de laboratorio en cada periodo.

El desarrollo del aplicativo será de gran utilidad debido a que los docentes involucrados tendrán acceso a una interfaz simple e intuitiva con el objetivo de solamente crear los ejercicios para las guías de laboratorio, permitiéndoles un control de las mismas a lo largo del tiempo. Adicionalmente el coordinador del claustro es el más interesado en dicho aplicativo, por ende, incentivará su uso a los miembros pertenecientes al mismo.

3.2 Análisis de requerimientos

En este presente subcapítulo se describe de forma detallada todos los requisitos y requerimientos de la herramienta colaborativa para prácticas de laboratorio del Claustro

Docente de Programación. El objetivo es analizar y entender las necesidades de los involucrados.

A continuación, se describirán las interfaces de usuario, los perfiles de usuario y sobre todo se especificará las funcionalidades y operaciones que debe realizar la aplicación.

3.2.1 Alcance

Este aplicativo tiene como objetivo proporcionar un ambiente de trabajo común entre los miembros pertenecientes al Claustro Docente de Programación, con el fin de administrar las prácticas de laboratorio. El aplicativo será desarrollado para la Universidad Politécnica Salesiana Sede Quito – Campus Sur

Concretamente, el aplicativo permitirá a los usuarios ingresar ejercicios para las guías de laboratorio determinadas por períodos académicos. Creadas por un coordinador de asignatura, los mismos se encuentran organizados dentro de carreras.

Adicionalmente, permitirá gestionar la información de los docentes que accedan al aplicativo, es decir se mantendrá la trazabilidad de los ejercicios propuestos por los docentes por cada periodo académico y la utilización de los mismos durante los diferentes períodos académicos.

El aplicativo no contempla su uso en otros claustros o carreras que no sean la carrera de ingeniería de sistemas y el claustro docente de programación de la Universidad Politécnica Salesiana, adicionalmente los ejercicios no podrán ser exportados a otras plataformas, ni mucho menos cambiar el formato de las guías de laboratorio previamente convenido.

3.2.2 Personal involucrado

El proyecto contempla la participación de 3 personas, la primera es el desarrollador backend a cargo de la base de datos y la comunicación y desarrollo de APIs. Ver Tabla 6.

Tabla 6

Persona involucrada

Definición	Funciones
Nombre	Jefferson Leonardo Oña Díaz
Rol	Programador Backend
Categoría profesional	Ingeniero en sistemas
Responsabilidades	Administrar la base de datos Desarrollar las APIs

Nota. Detalles de las personas involucradas en el desarrollo. Elaborado por: los autores.

El segundo involucrado es el desarrollador Frontend a cargo del desarrollo de la aplicación y de consumir las API's. Ver Tabla 7.

Tabla 7

Persona involucrada

Definición	Funciones
Nombre	Francisco Javier Cobo Valle
Rol	Programador Frontend
Categoría profesional	Ingeniero en sistemas
Responsabilidades	Diseñar las interfaces Consumir las APIs Desarrollar el aplicativo web

Nota. Detalles de las personas involucradas en el desarrollo. Elaborado por: los autores.

La tercera persona involucrada es el coordinador del proyecto, a cargo de darle seguimiento y orientar a los desarrolladores en la creación del mismo. Ver tabla 8.

Tabla 8

Persona involucrada

Definición	Funciones
Nombre	Tufiño Cárdenas Rodrigo Efraín
Rol	Seguimiento
Categoría profesional	Ingeniero en sistemas
Responsabilidades	Coordinador del proyecto

Nota. Detalles de las personas involucradas en el desarrollo. Elaborado por: los autores.

3.2.2 Definiciones, acrónimos y abreviaturas

Claustro docente: formado por docentes voluntarios, quienes comparten intereses académicos o investigación, que mediante el trabajo colaborativo crean una herramienta que desemboca en realización profesional. (Claustro Docente - UPS, 2017).

Coordinador de asignatura: persona encargada de la asignatura, el cual tiene como funciones generar las guías de laboratorio de cada periodo, coordinar el trabajo dentro del claustro docente, elaborar exámenes, convocar reuniones.

UPS: Universidad Politécnica Salesiana.

Interfaz: Vía por la cual el usuario interactúa con el sistema.

Aplicativo Web: Herramienta informática que utiliza un usuario para realizar tareas a través de un navegador Web.

Código Abierto: Software basado en la colaboración abierta para su desarrollo.

Base de datos: Es un lugar donde se almacena los datos que maneja el aplicativo web.

Frontend: La parte del aplicativo con la cual el usuario interactúa.

Backend: La parte del aplicativo encargado de toda la lógica de funcionamiento.

3.2.3 Descripción general

3.2.3.1. Perspectiva del producto. El aplicativo propuesto es independiente y funcionará de forma autónoma a otros sistemas existentes.

Demandará de una base de datos para almacenar la información y de un servidor de aplicaciones para su funcionamiento. Los usuarios accederán al aplicativo por medio de un navegador Web.

3.2.3.2. Funciones del producto. El sistema se lo desarrollará como una aplicación Web, los usuarios mediante un proceso de autenticación y dispondrá de los siguientes módulos:

- **Módulo de administración:** en este módulo se gestionará la información de usuarios, carreras y periodos.
- **Módulo de reportes:** Este módulo permitirá visualizar un registro de trazabilidad de los ejercicios propuestos, de acuerdo con las asignaturas que pertenezcan.
- **Módulo de ejercicios:** en este módulo los docentes podrán agregar nuevos ejercicios que más adelante serán añadidos en una guía de laboratorio, adicionalmente permitirá el ingreso de citas bibliográficas relacionadas al ejercicio, en el caso que sea necesario.
- **Módulo de guías:** en este módulo el coordinador de cada asignatura tiene la capacidad de generar una nueva guía de laboratorio, adicional podrá gestionar la viabilidad de cada ejercicio y cambiarlos de ser necesario hasta que decida dar por cerrada la guía de laboratorio.

3.2.3.3. Características de usuarios Se ha identificado tres tipos de usuarios en función de las actividades que pueden realizar dentro del sistema; estos usuarios son: Administrador y Docente, el detalle de sus actividades y descripción se encuentra adjunta en los anexos(características de usuarios).

3.2.3.4. Restricciones.

- La aplicación se desarrolló mediante el uso de software libre u open source por ende no se deberá pagar por el uso de: Sistema de Gestión de base de datos y el lenguaje de programación, por lo tanto, la utilización de este aplicativo se hará por medio de las políticas estipuladas por este tipo de licencia.

- Para este aplicativo se hará uso exclusivo de los navegadores Web mencionados a continuación Chrome, Opera, Microsoft Edge e Internet Explorer.
- El aplicativo solo podrá ser accedido por medio de una conexión a Internet.
- Todo el material destinado al usuario estará en lenguaje español.
- Los usuarios están limitados en las funciones que pueden realizar dentro del aplicativo de acuerdo con su rol.
- El sistema se diseñará según un esquema cliente-servidor.

3.2.3.5. Suposiciones y dependencias.

- Se asume que el usuario tiene acceso a una conexión de Internet para acceder al aplicativo.
- Se asume que los requisitos descritos en este documento son estables una vez que sea aprobado por las partes interesadas. Cualquier petición de cambios en la especificación debe ser aprobada por todas las partes interesadas.
- La disponibilidad del sistema dependerá de su alojamiento web, dada su arquitectura cliente-servidor.
- El aplicativo funciona independientemente, sin necesidad de comunicarse con otras aplicaciones externas, por lo que no hay dependencias con respecto a otros sistemas.

3.2.4 Requerimientos específicos

A continuación, se enlista los requerimientos específicos del aplicativo, a cada uno se le asignó una abreviación que consta de la letra R seguida de un número secuencial del 1 al 11 con el fin de identificarlos en el documento.

- R1: Gestión de usuario.
- R2: Autenticación de usuarios.
- R3: Gestión de periodos académicos.
- R4: Gestión de asignaturas.
- R5: Asignar coordinador de asignatura.
- R6: Designar docentes a la asignatura.
- R7: Asignar temas a la asignatura.
- R8: Asignar temas a la asignatura.
- R9: Gestión de ejercicios para las guías de laboratorio.
- R10: Gestión de guías de laboratorio.
- R11: Generación de reportes.

3.2.5 Requisitos funcionales

3.2.5.1. Requisito Funcional (RF1). El requisito 1 describe el proceso de gestión de usuarios, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 9.

Tabla 9

Requerimiento específico 1

Número	Nombre		Tipo	Prioridad
R1	Gestión de usuario		Requisito	Medio
Descripción	El sistema debe ser capaz de insertar, editar, mostrar y eliminar usuarios			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Cédula, Nombres, Apellidos, Correo, Contraseña, Rol	Mensaje de la acción realizada con éxito en pantalla	Mensaje de formulario llenado incorrectamente. Mensaje de error al realizar la acción	El sistema no debe permitir eliminar todos los usuarios administradores

Proceso	El administrador tiene acceso a una ventana con distintas opciones, en la cual podrá llenar los formularios de los usuarios para insertar, editar o eliminar, al momento de llenar el formulario correspondiente se verificará la validez de los datos ingresados y a continuación en caso de que no haya problemas se insertarán en la base de datos
Precondición	Tener un usuario administrador ya creado.

Nota. Detalles del requerimiento específico 1. Elaborado por: los autores.

3.2.5.2. Requisito Funcional (RF2). El requisito 2 describe el proceso de autenticación de usuarios, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 10.

Tabla 10

Requerimiento específico 2

Número	Nombre		Tipo	Prioridad
R2	Autenticación de usuarios.		Requisito	Medio
Descripción	Ingresar en el sistema con una cuenta de usuario			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Correo, Contraseña	Mensaje de la acción realizada con éxito en pantalla	Mensaje de formulario llenado incorrectamente. Mensaje de usuario o contraseña incorrectos	El requisito R2 no cuenta con ninguna restricción
Proceso	Cualquier usuario al momento de ingresar al sistema, si no tiene la sesión iniciada se desplegará un formulario para ingresar sus datos de correo y contraseña, si estos son correctos ingresa a la vista principal caso contrario seguirá en el formulario.			
Precondición	Tener una cuenta de usuario ya creada con su correo y contraseña			

Nota. Detalles del requerimiento específico 2. Elaborado por: los autores.

3.2.5.3. Requisito Funcional (RF3). El requisito 3 describe el proceso de Gestión de periodos académicos, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 11.

Tabla 11

Requerimiento específico 3

Número	Nombre		Tipo	Prioridad
R3	Gestión de periodos académicos.		Requisito	Medio
Descripción	Iniciar un nuevo periodo académico.			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Número del periodo, fecha de inicio, fecha de fin, estado.	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción	Únicamente un periodo puede estar activo
Proceso	El usuario administrador tiene acceso a un apartado en donde podrá modificar el periodo con el que se generarán las guías de laboratorio.			
Precondición	Este requisito no tiene ninguna precondición			

Nota. Detalles del requerimiento específico 3. Elaborado por: los autores.

3.2.5.4. Requisito Funcional (RF4). El requisito 4 describe el proceso de Gestión de asignaturas, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 12.

Tabla 12

Requerimiento específico 4

Número	Nombre		Tipo	Prioridad
R4	Gestión de asignaturas.		Requisito	Alto
Descripción	El sistema debe ser capaz de insertar, editar, mostrar, eliminar			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Código de asignatura, nombre, coordinador, carrera	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	El requisito R4 no cuenta con ninguna restricción
Proceso	El administrador tiene acceso a una ventana con distintas opciones, en la cual podrá llenar los formularios de las asignaturas para insertar, editar o eliminar, al momento de llenar el formulario correspondiente se verificará la validez de los datos ingresados y a continuación en caso de que no haya problemas se insertarán en la base de datos			
Precondición	Tener un usuario con el rol de docente ya creado			

Nota. Detalles del requerimiento específico 4. Elaborado por: los autores.

3.2.5.5. Requisito Funcional (RF5). El requisito 5 describe el proceso de asignar coordinador de asignatura , detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 13.

Tabla 13

Requerimiento específico 5

Número	Nombre		Tipo	Prioridad
R5	Asignar coordinador de asignatura		Requisito	Alto
Descripción	El sistema debe ser capaz de asignar un coordinador a una asignatura			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Coordinador de la asignatura	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	El requisito R5 no cuenta con ninguna restricción
Proceso	El administrador tiene acceso a una ventana con distintas opciones, en la cual podrá llenar los formularios de las asignaturas para asignar a un coordinador de una asignatura, al momento de llenar el formulario correspondiente se verificará la validez de los datos ingresados y a continuación en caso de que no haya problemas se insertarán en la base de datos			
Precondición	Tener un usuario con el rol de docente ya creado			

Nota. Detalles del requerimiento específico 5. Elaborado por: los autores.

3.2.5.6. Requisito Funcional (RF6). El requisito 6 describe el proceso de añadir docentes a la asignatura, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 14.

Tabla 14

Requerimiento específico 6

Número	Nombre		Tipo	Prioridad
R6	Designar docentes a la asignatura.		Requisito	Alto
Descripción	Agregar información adicional de una asignatura por el coordinador			

Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Docentes.	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	Únicamente los coordinadores de su asignatura podrán agregar docentes a su asignatura
Proceso	Los coordinadores de cada materia tienen acceso a una vista donde se podrá modificar los docentes que pertenecen a la asignatura.			
Precondición	Tener asignaturas creadas			

Nota. Detalles del requerimiento específico 6. Elaborado por: los autores.

3.2.5.7. Requisito Funcional (RF7). El requisito 7 describe el proceso de añadir temas a la asignatura, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 15.

Tabla 15

Requerimiento específico 7

Número	Nombre		Tipo	Prioridad
R7	Asignar temas a la asignatura.		Requisito	Alto
Descripción	Agregar información adicional de una asignatura por el coordinador			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Temas, subtemas.	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	Únicamente los coordinadores de su asignatura podrán agregar temas a su asignatura
Proceso	Los coordinadores de cada materia tienen acceso a una vista donde se podrá modificar los temas que tiene cada materia en caso de que el tema no exista se creará			
Precondición	Tener asignaturas creadas			

Nota. Detalles del requerimiento específico 7. Elaborado por: los autores.

3.2.5.8. Requisito Funcional (RF8). El requisito 8 describe el proceso de gestión de referencias bibliográficas, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 16.

Tabla 16

Requerimiento específico 8

Número	Nombre		Tipo	Prioridad
R8	Gestión referencias bibliográficas.		Requisito	Medio
Descripción	El sistema debe ser capaz de insertar, editar, mostrar, eliminar referencias			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Fuente, Referencia, Autor	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	El requisito R8 no cuenta con ninguna restricción
Proceso	El docente tiene acceso a una ventana con distintas opciones, en la cual podrá llenar los formularios de las referencias para insertar, editar o eliminar, al momento de llenar el formulario correspondiente se verificará la validez de los datos ingresados y a continuación en caso de que no haya problemas se insertarán en la base de datos			
Precondición	Este requisito no tiene ninguna precondición			

Nota. Detalles del requerimiento específico 8. Elaborado por: los autores.

3.2.5.9. Requisito Funcional (RF9). El requisito 9 describe el proceso de gestión de ejercicios, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 17.

Tabla 17

Requerimiento específico 9

Número	Nombre		Tipo	Prioridad
R9	Gestión de ejercicios para las guías de laboratorio.		Requisito	Alto
Descripción	El sistema debe ser capaz de insertar, editar, mostrar, eliminar, y asignar referencias a un ejercicio			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Nombre, descripción corta, descripción larga, dificultad, ejercicio, ejemplo	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	Los ejercicios ya creados solo pueden ser modificados por el docente que lo creo o el coordinador
Proceso	El Docente tiene una ventana con distintas opciones, en la cual podrá llenar los formularios de un ejercicio para insertar, editar o eliminar, al momento de llenar el formulario correspondiente se verificará la validez de los datos ingresados y a continuación en caso de que no haya problemas se insertarán en la base de datos si no existe la referencia.			
Precondición	El Docente debe pertenecer a la asignatura a la cual quiere gestionar un ejercicio			

Nota. Detalles del requerimiento específico 9. Elaborado por: los autores.

3.2.5.10. Requisito Funcional (RF10). El requisito 10 describe el proceso de gestión de guías de laboratorio, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 18.

Tabla 18

Requerimiento específico 10

Número	Nombre		Tipo	Prioridad
R10	Gestión de guías de laboratorio.		Requisito	Alto
Descripción	El sistema debe ser capaz de insertar, editar, mostrar, eliminar, y asignar ejercicios a una guía de laboratorio			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	asignatura, periodo, formato, título, objetivo, requisitos, instrucciones, resultados, conclusiones, actividades,	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción. Mensaje de formulario llenado incorrectamente.	El requisito R5 no cuenta con ninguna restricción
Proceso	Si el coordinador de la asignatura desea crear una guía de laboratorio tiene la opción de generar los ejercicios escogiendo los temas, en caso de los ejercicios generados aleatoriamente no sean del agrado del coordinador se podrá cambiar y una vez esté conforme podrá llenar la información de la práctica para cerrar la guía de laboratorio y subir esa información a la base de datos			
Precondición	Tener ejercicios ya creados			

Nota. Detalles del requerimiento específico 10. Elaborado por: los autores.

3.2.5.9. Requisito Funcional (RF11). El requisito 11 describe el proceso de generar reportes, detallando el proceso, la entrada y salida de datos, si existe alguna restricción o si requiere de una pre condición para su ejecución. Ver Tabla 19.

Tabla 19

Requerimiento específico 11

Número	Nombre	Tipo	Prioridad
R11	Generación de reportes.	Requisito	Medio

Descripción	Generar reportes que ayuden a visualizar la información de mejor forma			
Fuente	Entrada	Salida exitosa	Salida conflicto	Restricción
Coordinador Claustro Docente	Este requisito no cuenta con una entrada definida	Mensaje de la acción realizada con éxito en pantalla	Mensaje de error al realizar la acción en pantalla	Los reportes solo deben ser generados por usuarios con permisos en cada tipo de reporte
Proceso	El usuario tiene a disposición una vista solo para generar el reporte al cual tiene permisos designados de generar, en caso de que no tenga el permiso no le aparecerá			
Precondición	Información del reporte que se desee generar			

Nota. Detalles del requerimiento específico 11. Elaborado por: los autores.

3.2.6 Requisitos no funcionales

3.2.6.1. Requisitos de rendimiento.

- El aplicativo estará en línea las 24 horas del día los 7 días de la semana.
- Deberá soportar por lo menos 25 usuarios concurrente.
- Se estima un pico de transacciones al inicio de cada periodo académico.

3.2.6.2. Seguridad. El sistema de seguridad está basado en el uso de contraseñas para cada usuario, con el fin de que solo los usuarios autorizados tengan acceso, y definir un tiempo límite de inactividad después del inicio de sesión de los usuarios.

Las contraseñas de los usuarios se almacenan encriptadas.

3.2.6.3. Disponibilidad. El aplicativo ha sido diseñado tomando en cuenta las necesidades de los involucrados, por lo que se encuentra disponible el 80% del día, es decir 18 horas al día, mientras el 20% restante es dedicado a tareas administrativas del aplicativo.

3.2.6.4. Escalabilidad. El aplicativo está previsto para funcionar en cualquier carrera y materia de la Universidad Politécnica Salesiana, su uso no es exclusivo del Claustro Docente de programación.

3.3 Diseño

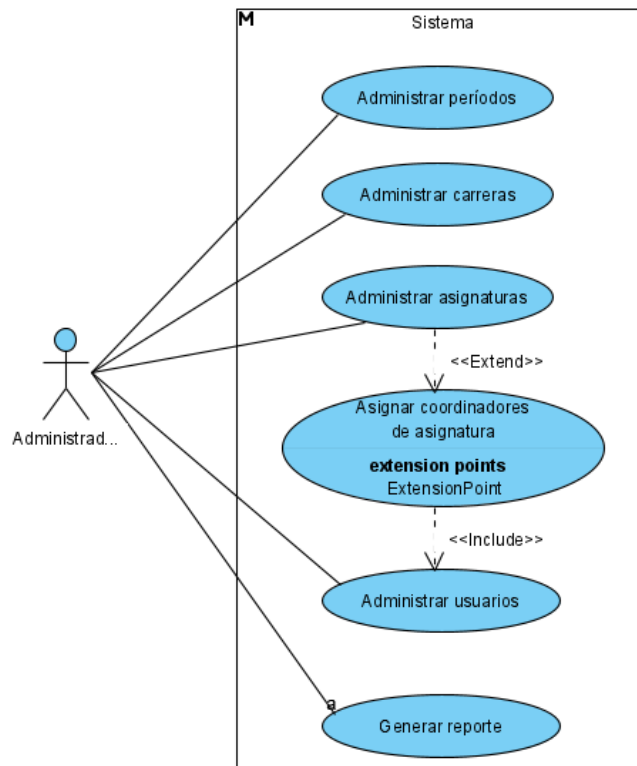
El diseño del aplicativo está representado por medio de diagramas UML, con el objetivo de definir el sistema a construir, se realizó los siguientes diagramas.

3.3.1 Diagramas de casos de uso

En la figura No. 3 se puede observar como el usuario con perfil administrador, puede interactuar con el aplicativo, las acciones que puede ejecutar son: administrar periodos, administrar carreras, administrar asignaturas que extiende sus funciones a la asignación de un coordinador de asignatura la cual incluye la administración de usuarios y finalmente la generación de reporte.

Figura 3

Caso de uso para usuario Administrador

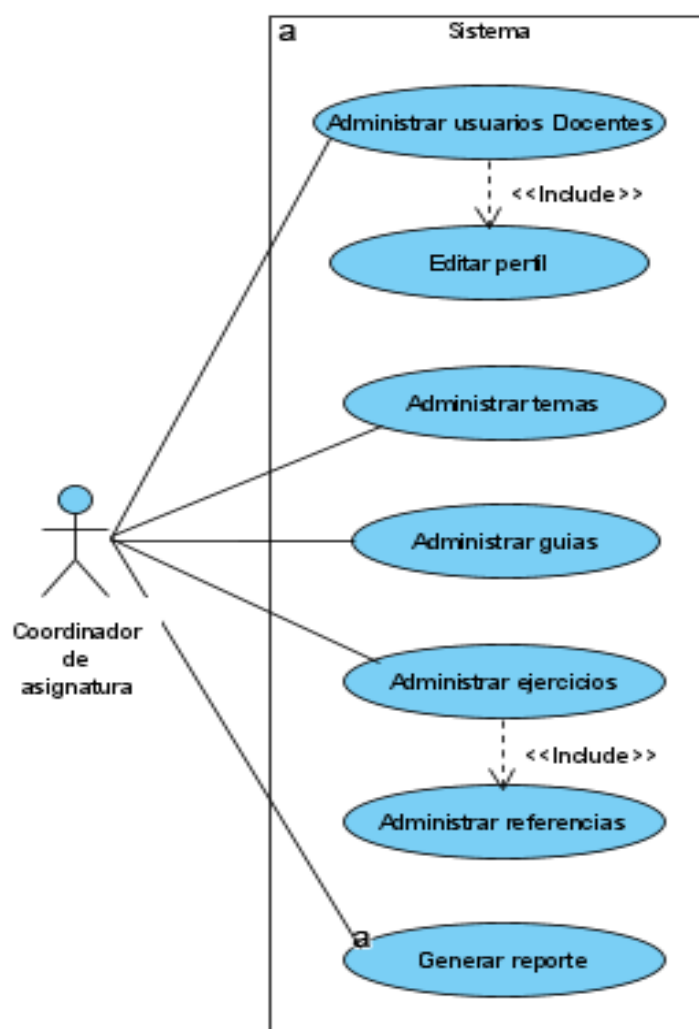


Nota. Actividades que puede realizar el Administrador Elaborado por: los autores.

En la figura No. 4 se puede observar como el usuario con perfil coordinador de asignatura, puede interactuar con el aplicativo, las acciones que puede ejecutar son: administrar usuarios solo de tipo Docente la cual incluye la edición del perfil, administrar los temas de la asignatura, administrar las guías de laboratorio, administrar los ejercicios el cual incluye la administración de citas bibliográficas y finalmente la generación de reportes.

Figura 4

Caso de uso para usuario Coordinador de asignatura

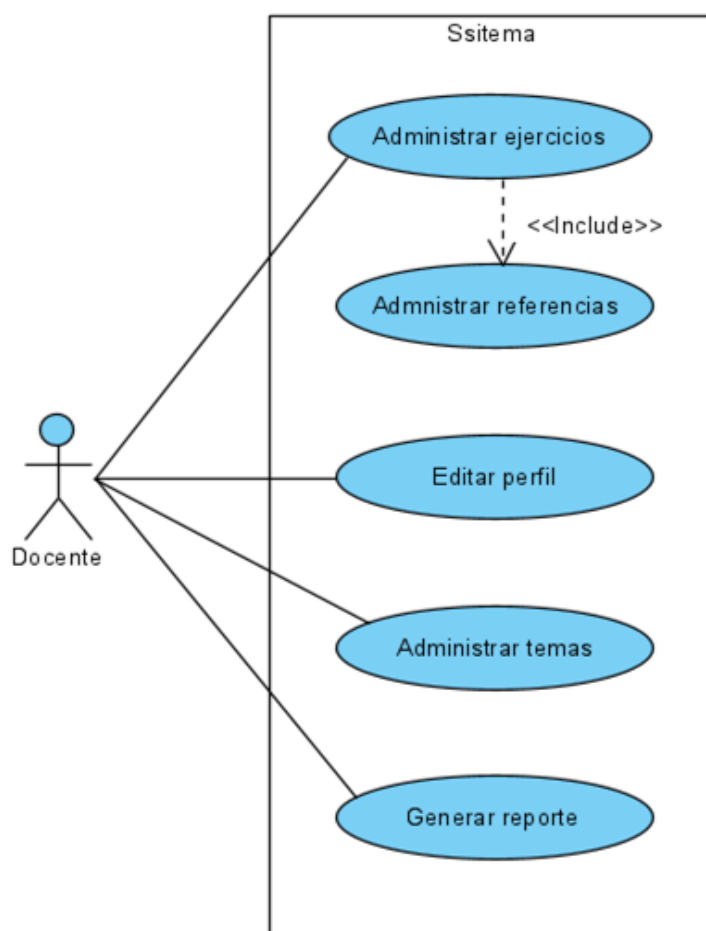


Nota. Actividades que puede realizar el Coordinador de asignatura. Elaborado por: los autores.

En la figura No. 5 se puede observar como el usuario con perfil docente, puede interactuar con el aplicativo, las acciones que puede ejecutar son: administrar los ejercicios el cual incluye la administración de citas bibliográficas, editar su propio perfil de usuario, administrar temas de sus asignaturas y finalmente la generación de reportes.

Figura 5

Caso de uso para usuario Docente



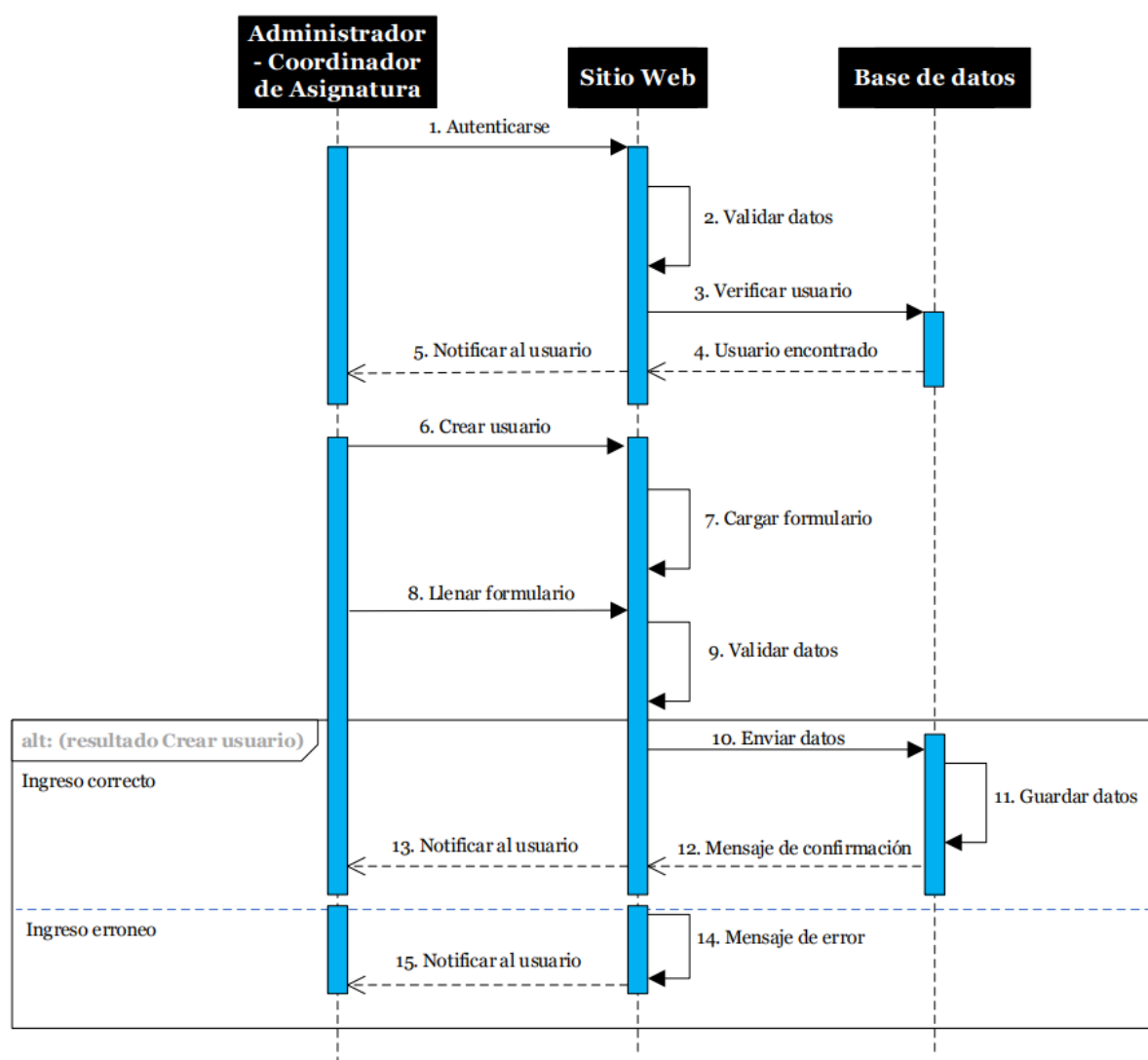
Nota. Actividades que puede realizar el Docente. Elaborado por: los autores.

3.3.2 Diagramas de secuencia

En la figura No. 6 se puede observar la interacción entre Administrador/Coordinador de asignatura, sitio web y base de datos en el proceso de creación de usuarios.

Figura 6

Diagrama de secuencia: Crear usuario

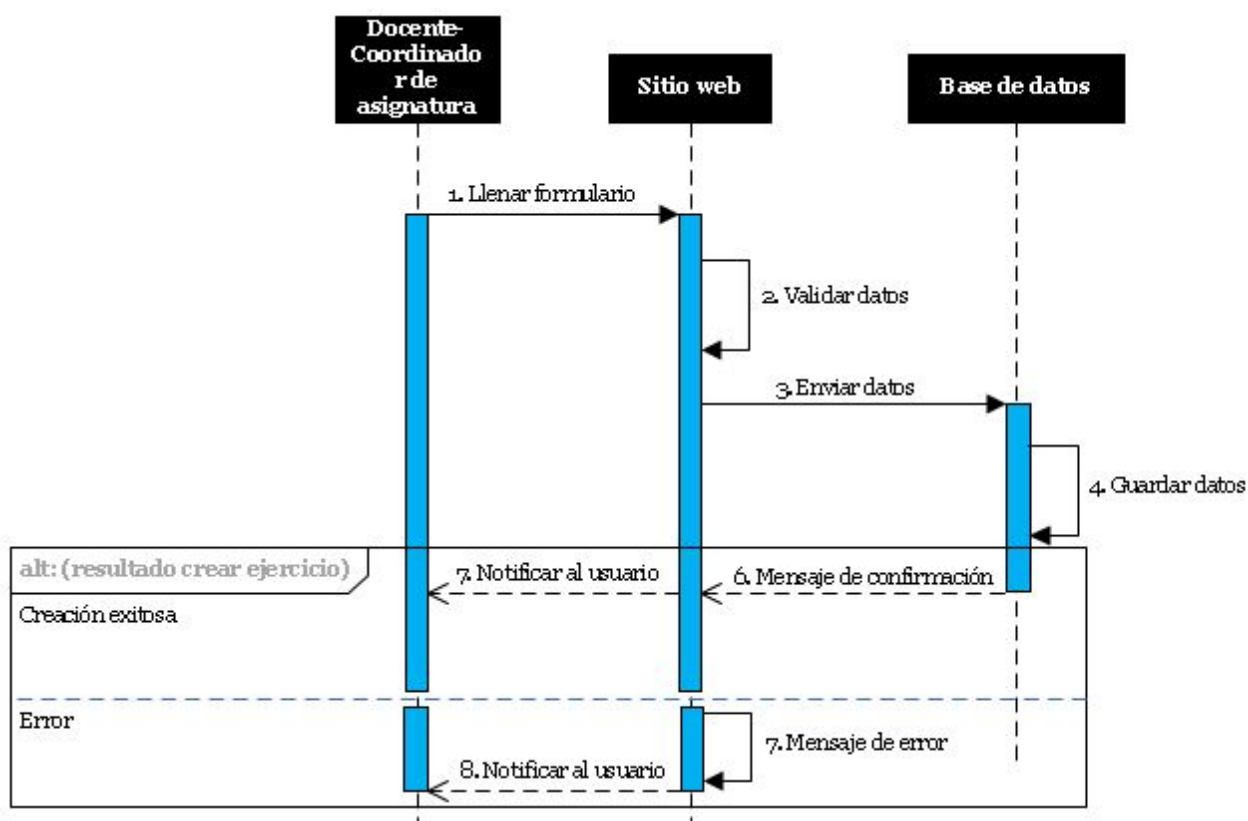


Nota. Diagrama de secuencia del proceso de creación de nuevos usuarios, por parte del usuario administrador y coordinador de asignatura. Elaborado por: los autores.

En la figura No. 7 se puede observar la interacción entre Docente/Coordinador de asignatura, sitio web y base de datos en el proceso de creación de ejercicios para las guías de laboratorio.

Figura 7

Diagrama de secuencia: Crear Ejercicio

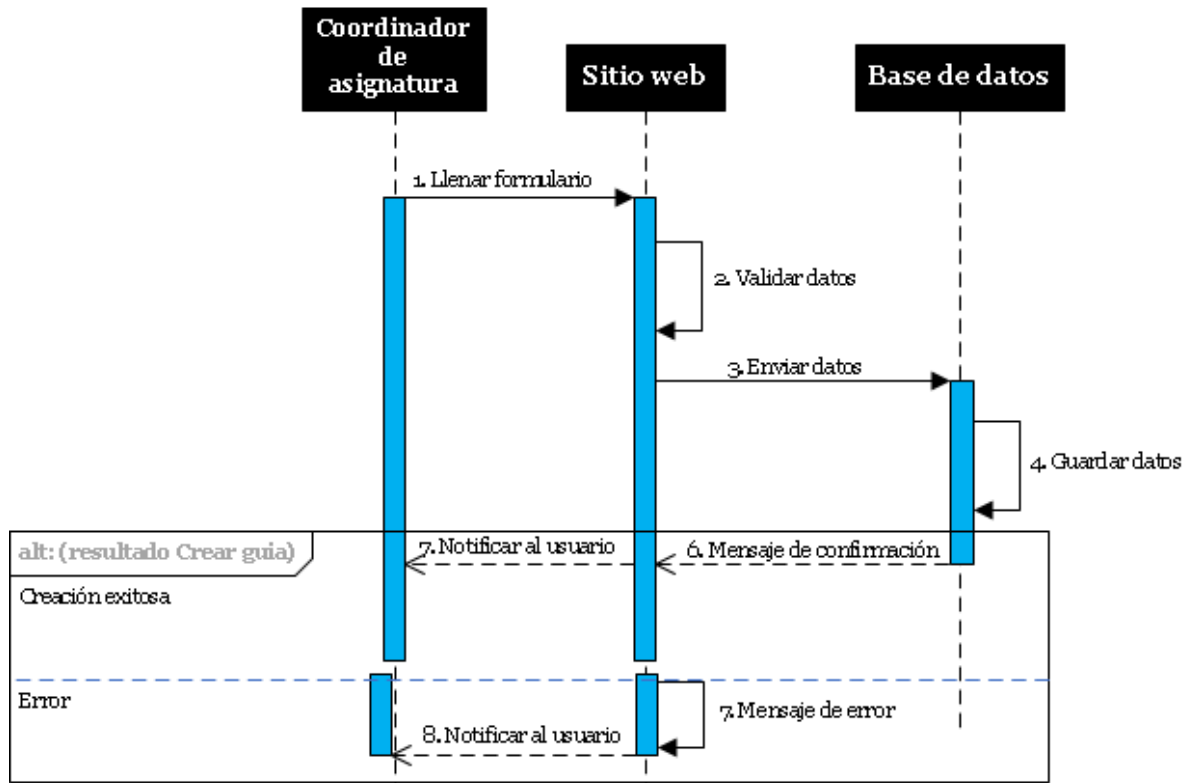


Nota. Diagrama de secuencia del proceso de creación de ejercicios para las guías de laboratorio, por parte de los usuarios Docentes y Coordinadores de asignatura. Elaborado por: los autores.

En la figura No. 8 se puede observar la interacción entre Coordinador de asignatura, sitio web y base de datos en el proceso de crear la guía de laboratorio, crearla hace referencia a llenar los datos informativos de la guía mas no los ejercicios que la conforman.

Figura 8

Diagrama de secuencia: Crear guía de laboratorio

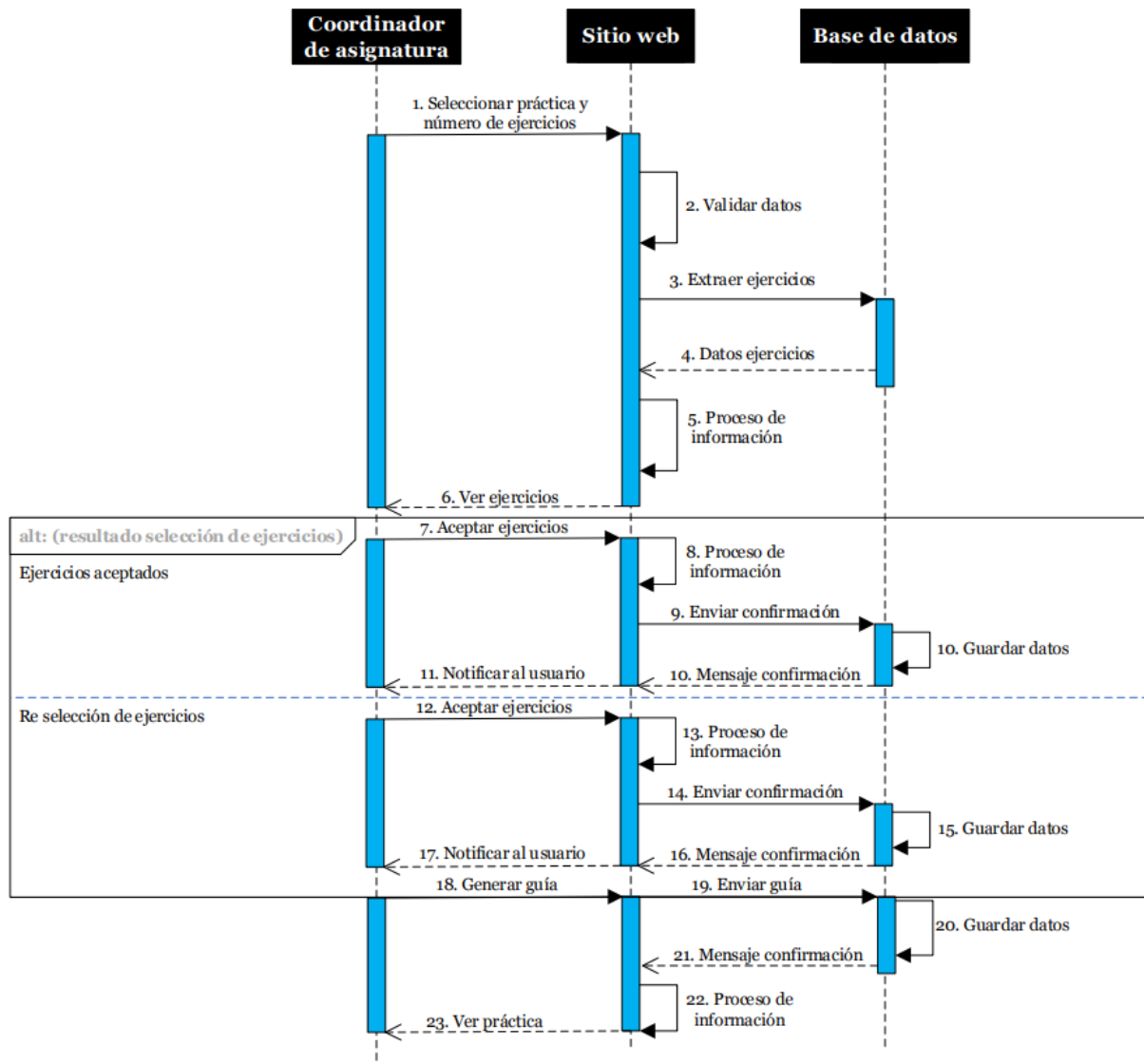


Nota. Diagrama de secuencia del proceso de creación de guías de laboratorio, por parte de los usuarios Coordinadores de asignatura. Elaborado por: los autores.

En la figura No. 9 se puede observar la interacción entre Coordinador de asignatura, sitio web y base de datos en el proceso de generar la guía de laboratorio, en este punto es donde los ejercicios son asignados a la guía de laboratorio y se da por completa.

Figura 9

Diagrama de secuencia: Generar guía de laboratorio

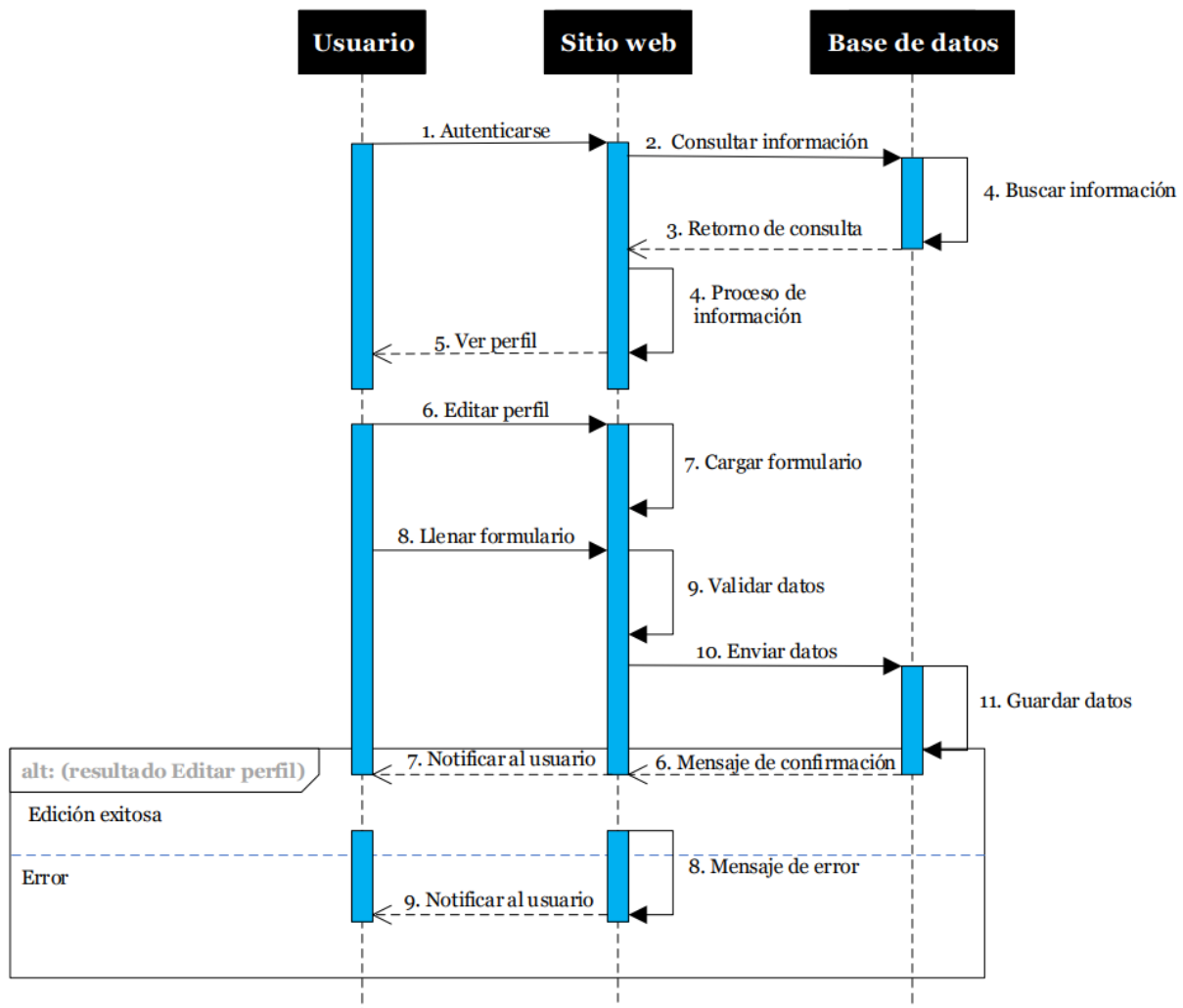


Nota. Diagrama de secuencia del proceso de generación de guías de laboratorio, por parte del usuario Coordinador de asignatura. Elaborado por: los autores.

En la figura No. 10 se puede observar la interacción entre cualquier usuario del sistema, sitio web y base de datos en el proceso de editar la información del perfil del usuario.

Figura 10

Diagrama de secuencia: Editar perfil



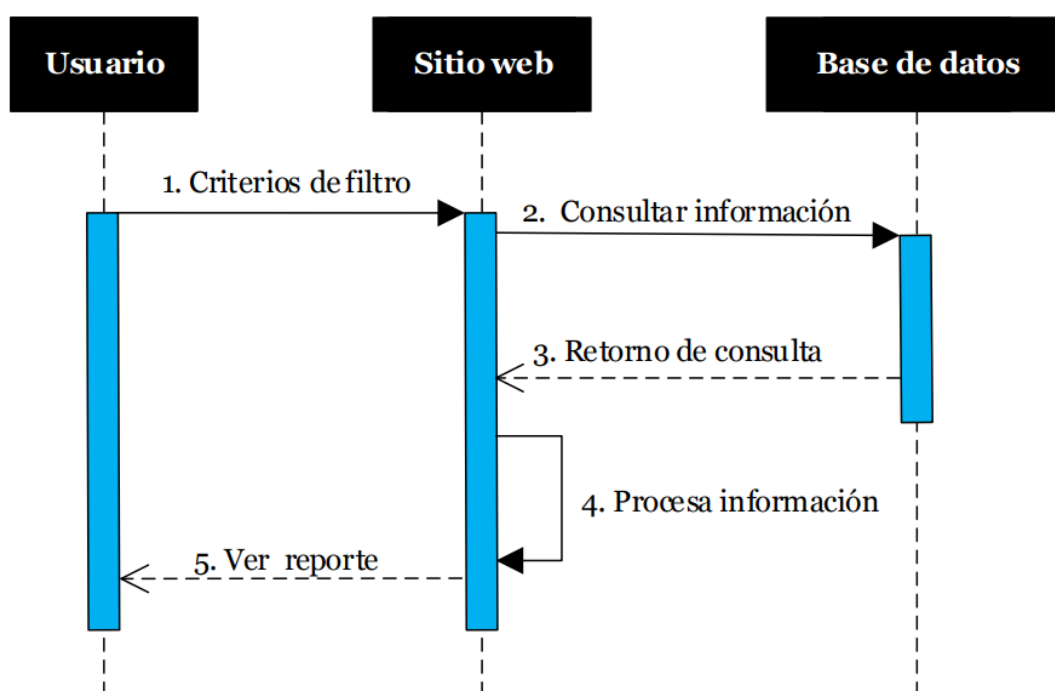
Nota. Diagrama de secuencia del proceso de editar perfil, por parte del cualquier usuario.

Elaborado por: los autores.

En la figura No. 11 se puede observar la interacción entre cualquier usuario del sistema, sitio web y base de datos en el proceso de generar reporte de trazabilidad de acuerdo con la selección de filtros del usuario.

Figura 11

Diagrama de secuencia: Generar reporte



Nota. Diagrama de secuencia del proceso de generación de reporte, por parte del cualquier usuario. Elaborado por: los autores.

3.3.3. Diagrama conceptual de base de datos

El diagrama conceptual se puede ver desde 3 puntos de vista. Empezando por las asignaturas una de las principales tablas, en donde se guarda información importante como los docentes que pertenecen a esa asignatura así relacionándose con la tabla de docentes, pero a su vez la asignatura necesita un coordinador de asignatura por lo cual también cuenta con una relación con la tabla de coordinador. Para finalizar en las asignaturas también se guardará

información adicional como los temas que se dictan en dicha asignatura juntos a las referencias bibliográficas.

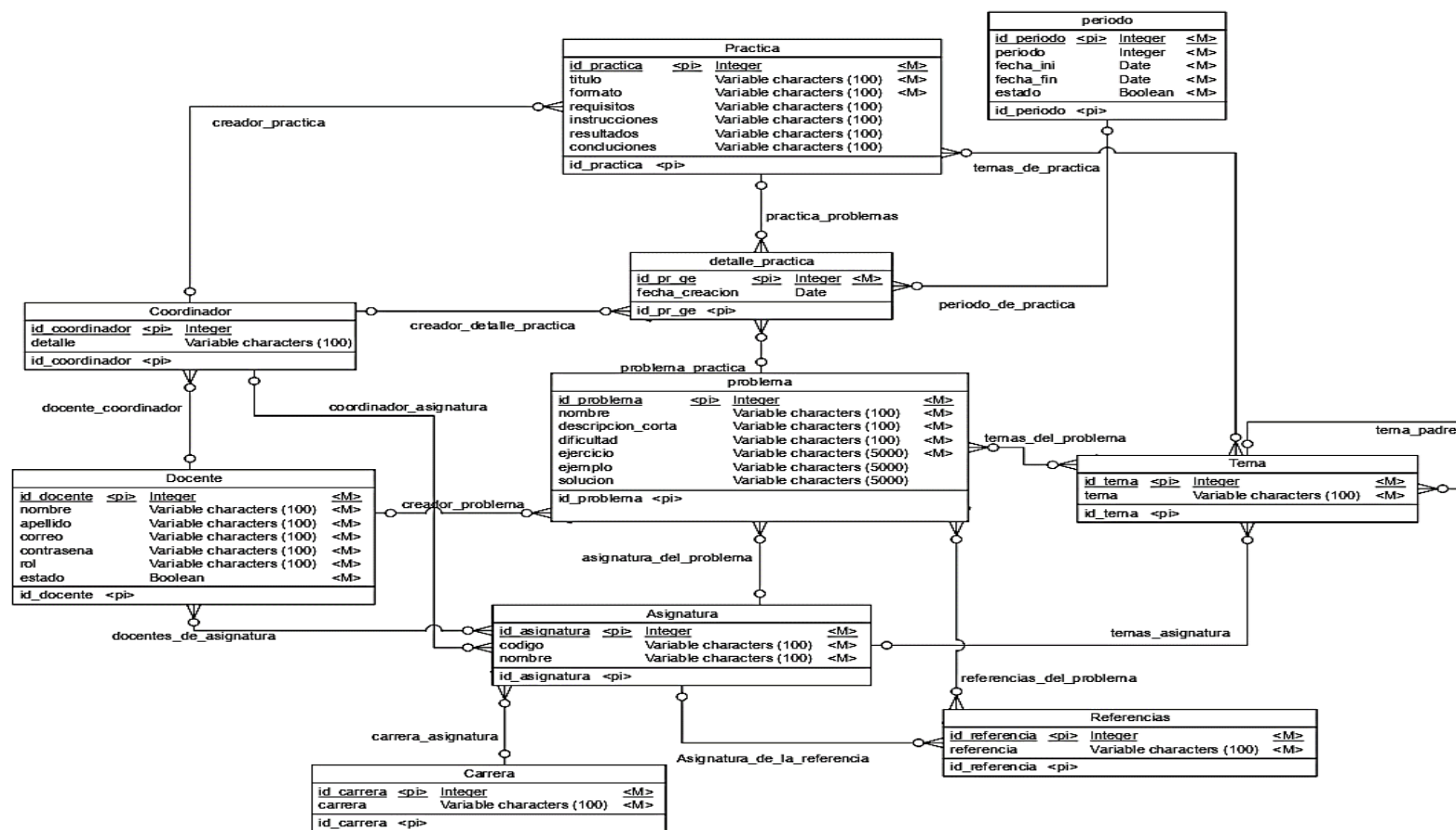
La tabla de prácticas servirá para generar un estándar de prácticas que serán reutilizadas en cada periodo, estas solo podrán ser generadas por un coordinador de asignatura y tendrán temas los cuales servirán para que en la tabla de detalle práctica poder generar los ejercicios aleatorios en base a esos temas.

En la tabla de problemas es donde se almacenará los ejercicios que tendrá la práctica de forma individual serán creados por un docente y además contiene sus temas, los cuales también servirán para la tabla de detalle práctica y generar las guías de laboratorio por periodo académico.

En la figura 12 se representó el diagrama conceptual de la base de datos de manera relacional con el fin de tener una estructura entendible, a pesar de que la base de datos es no relacional.

Figura 12

Diagrama conceptual de base de datos



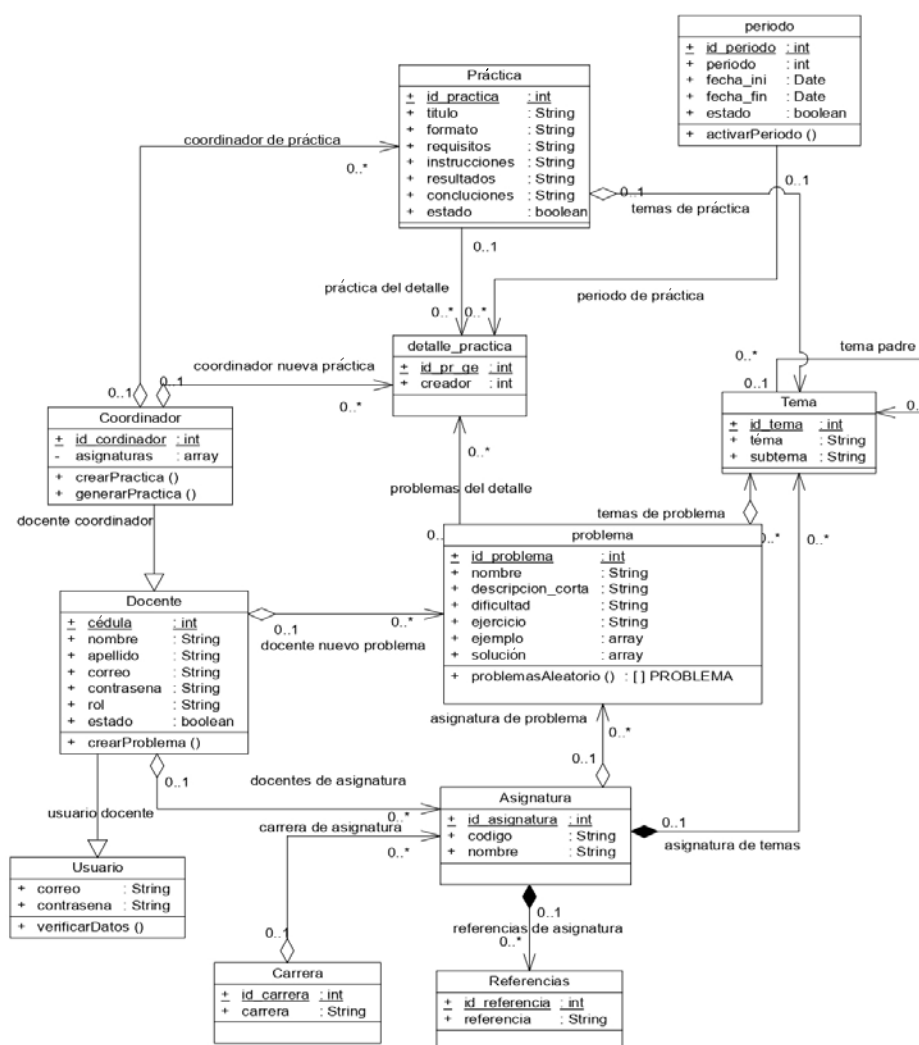
Nota. Diagrama donde se muestran las tablas y sus relaciones de la estructura de base de datos. Elaborado por: los autores.

3.3.4 Diagrama de clases

En el diagrama de clases se resalta el cómo los componentes de cada clase van a interactuar entre sí, lo más destacable es como las clases de referencia y temas tienen una relación de composición con asignatura debido a que en caso de eliminación de una asignatura consecuentemente también se eliminará las referencias y los temas de esa asignatura. A diferencia de la clase docentes, que, en caso de eliminación, los docentes de esa asignatura seguirán sin cambios gracias a la relación de agregación.

Figura 13

Diagrama de clases de todo el aplicativo



Nota. Diagrama de las clases y dependencias en el aplicativo. Elaborado por: los autores.

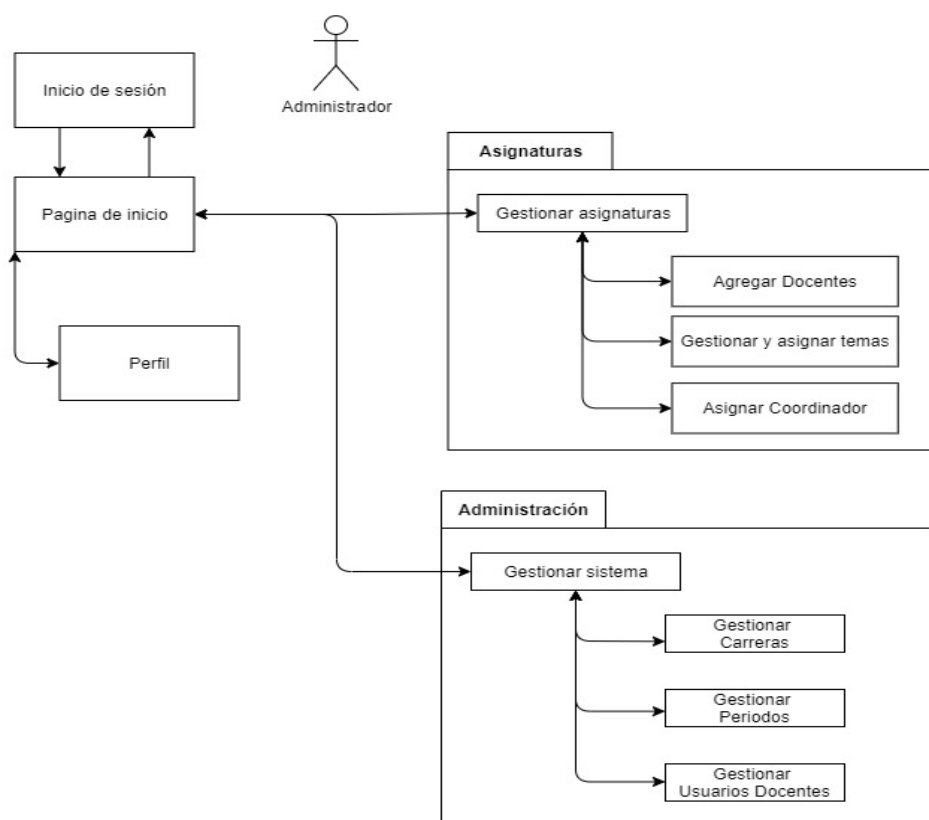
3.3.5 Mapas de navegación

El administrador está a cargo de 2 módulos importantes, en el primero módulo de asignaturas el administrador es el único tipo de usuario que podrá crear nuevas asignaturas además de agregar a su respectivo coordinador de la asignatura, el administrador también en caso de ser necesario podrá agregar docentes a las asignaturas.

En el módulo de administración este tipo de usuario estará a cargo de iniciar nuevos periodos académicos además de agregar más usuarios administradores y agregar nuevas carreras.

Figura 14

Mapa de navegación: Administrador



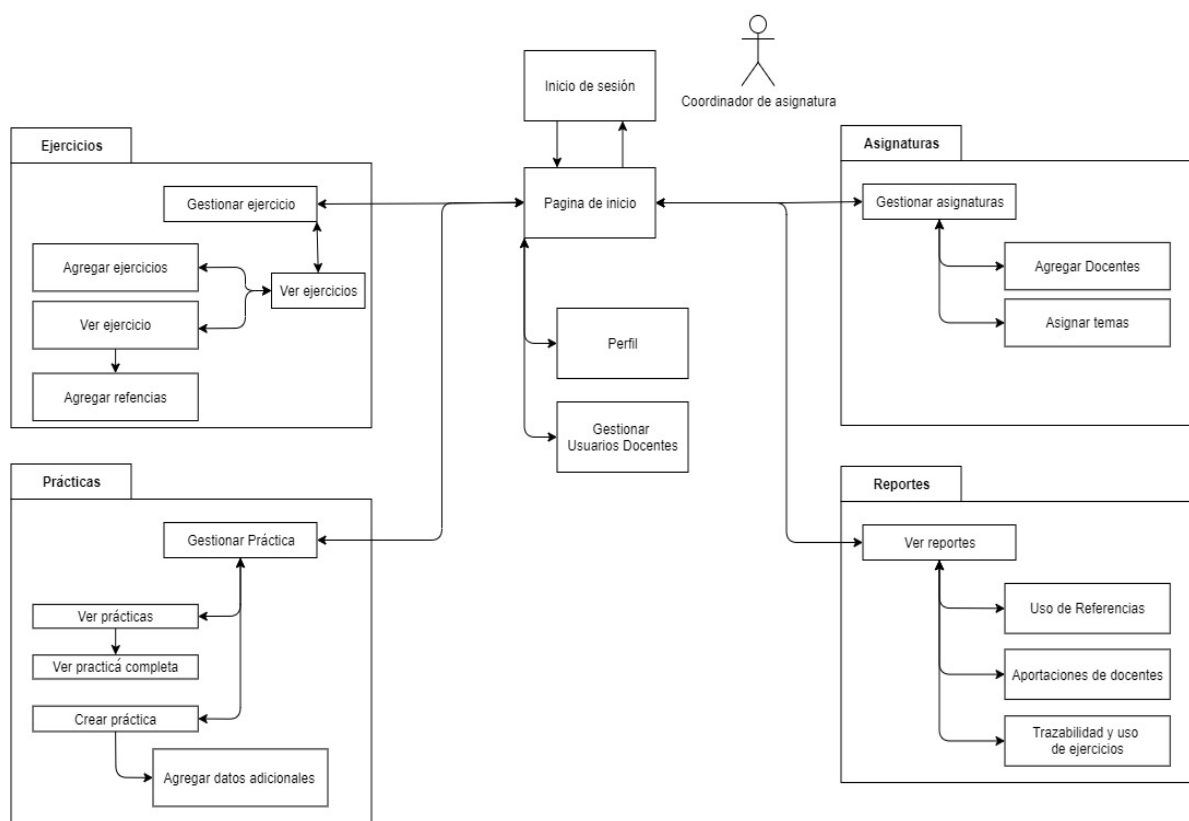
Nota. Mapa de navegación entre las páginas del aplicativo que tiene acceso el usuario de tipo administrador. Elaborado por: los autores.

Los coordinadores de asignatura son los que tienen más accesos a funciones del aplicativo, comenzando con lo único que puede hacer un usuario con ese rol, crear nuevas prácticas y proveerlas de ejercicios para cada periodo.

Los coordinadores de asignatura también pueden agregar nuevos usuarios con el rol docente y a su vez agregar esos usuarios a la asignatura que coordina adicional los coordinadores de asignatura tendrán acceso a todos los reportes del aplicativo.

Figura 15

Mapa de navegación: Coordinador de asignatura

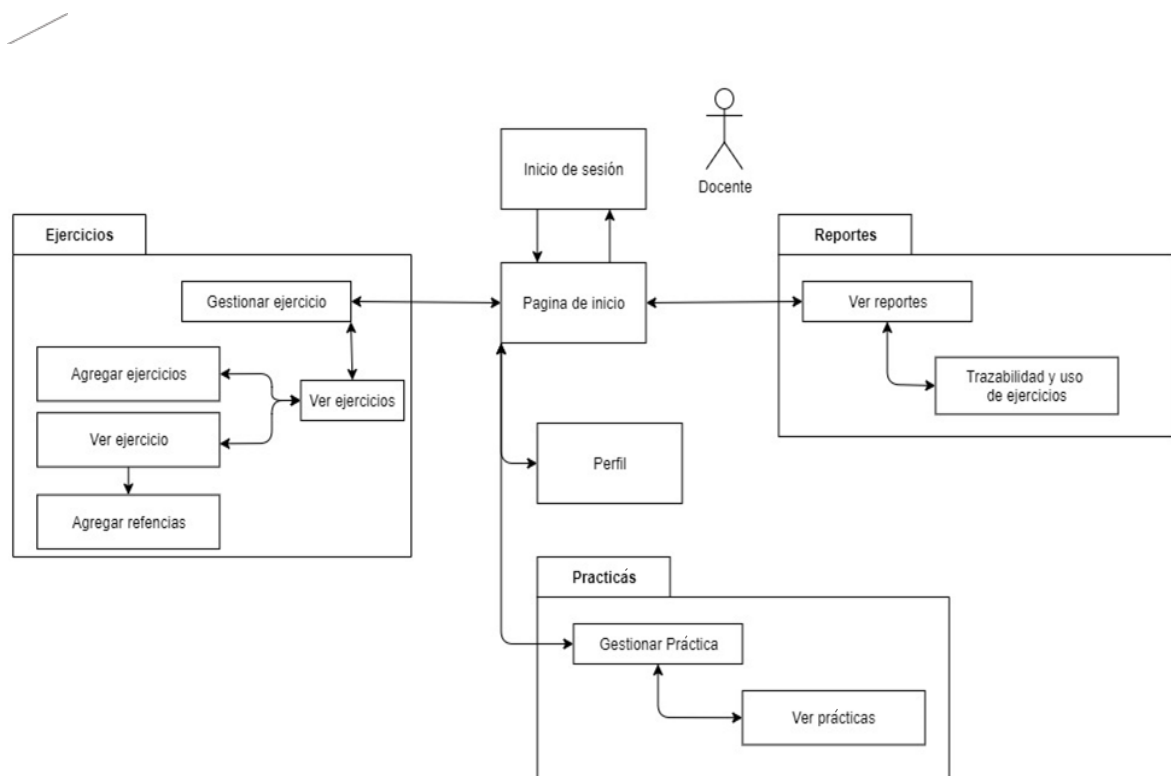


Nota. Mapa de navegación entre las páginas del aplicativo que tiene acceso el usuario coordinador de asignatura. Elaborado por: los autores.

A diferencia de los coordinadores de asignatura los docentes no tienen muchas funciones en el aplicativo, pueden agregar ejercicios con sus respectivas referencias además de solo ver las guías de laboratorio generadas y ver un solo tipo de reporte.

Figura 16

Mapa de navegación: Docente



Nota. Mapa de navegación entre las páginas del aplicativo que tiene acceso el usuario de tipo docente. Elaborado por: los autores.

3.3.6 Interfaces abstractas

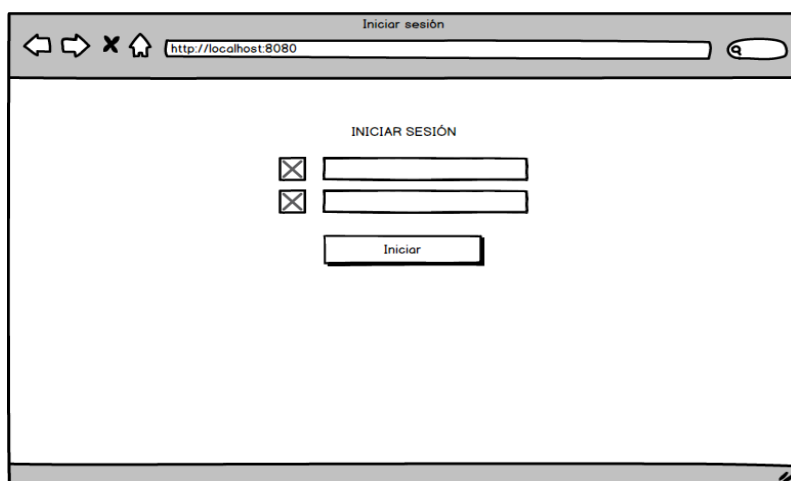
El diseño de las interfaces del aplicativo está enfocado desde los 3 tipos de usuario que existen: administrador, docente y coordinador de asignatura.

3.3.6.1. Interfaces abstractas comunes para todos los usuarios Interfaz abstracta

donde el usuario ingresará sus credenciales y podrá iniciar sesión en el aplicativo.

Figura 17

Interfaz abstracta: inicio de sesión



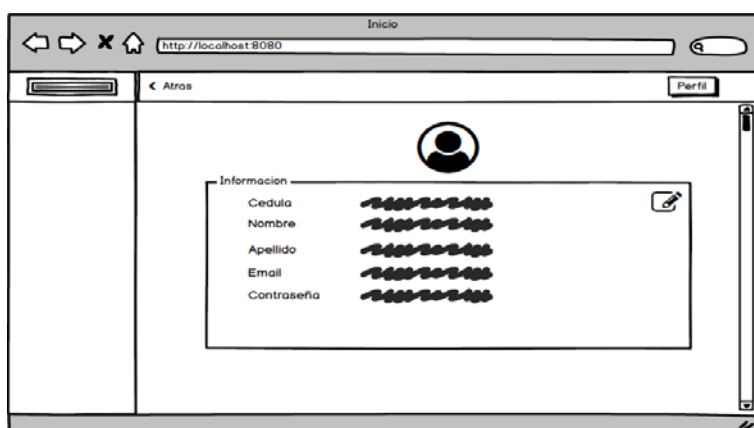
The diagram shows a web browser window titled "Iniciar sesión" with the address bar set to "http://localhost:8080". The main content area is titled "INICIAR SESIÓN" and contains two input fields, each preceded by a small square icon with an 'X'. Below these fields is a button labeled "Iniciar".

Nota. Diseño abstracto para iniciar sesión en el sistema. Elaborado por: los autores.

Interfaz para el usuario que haya iniciado sesión pueda ver su información personal y en caso de ser necesario también podrá editar dicha información.

Figura 18

Interfaz abstracta: perfil de usuario



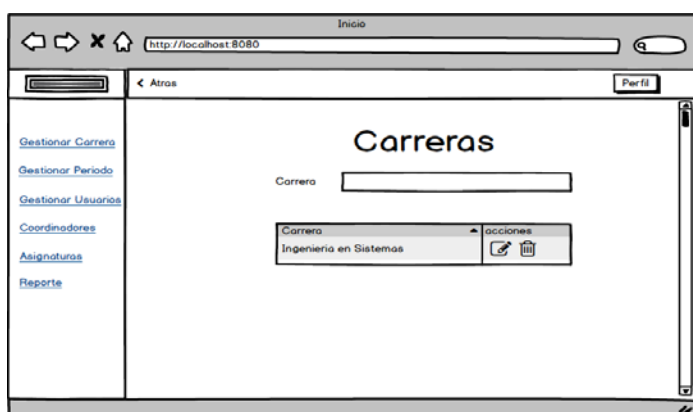
The diagram shows a web browser window titled "Inicio" with the address bar set to "http://localhost:8080". The page has a navigation bar with a "Perfil" button and a "Atras" button. The main content area features a user profile card with a circular profile picture icon. Below the icon, the card is titled "Información" and lists five fields: "Cedula", "Nombre", "Apellido", "Email", and "Contraseña". Each field is represented by a series of asterisks, indicating masked text. An edit icon (a pencil inside a square) is located to the right of the "Contraseña" field.

Nota. Interfaz abstracta para ver y editar el perfil del usuario. Elaborado por: los autores.

3.3.6.2. Interfaces abstractas del Administrador. Interfaz del administrador para ingresar en el sistema nuevas carreras que más adelante servirán para agregarlas en una asignatura.

Figura 19

Interfaz abstracta: gestionar carreras

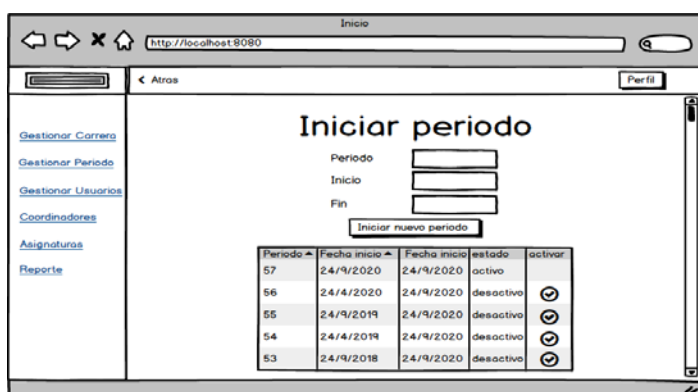


Nota. Interfaz abstracta para crear editar y eliminar carreras. Elaborado por: los autores.

Interfaz donde el usuario administrador tiene acceso a ingresar nuevos periodos académicos. Cuando empiece un nuevo periodo académico el administrador puede activar el periodo desde la misma tabla y a su vez desactivará el periodo anterior.

Figura 20

Interfaz abstracta: gestionar periodos



Nota. Interfaz abstracta para agregar y activar un periodo académico. Elaborado por: los autores.

Interfaz del administrador para gestionar asignaturas donde se podrá agregar, modificar, eliminar asignaturas y a su vez también agregar la carrera a la que pertenece.

Figura 21

Interfaz abstracta: gestionar asignaturas

The screenshot shows a web browser window with the URL `http://localhost:8080`. The page title is "Gestionar Asignaturas". On the left is a sidebar with links: "Gestionar Carrera", "Gestionar Periodo", "Gestionar Usuarios", "Coordinadores", "Asignaturas", and "Reporte". The main content area has a form titled "Ingresar usuario" with fields for "Código", "Nombre", and "Carrera" (a dropdown menu). Below the form is an "Aceptar" button. At the bottom, there is a table with the following data:

Código	Nombre	Carrera	Coordinador	acciones
26031	Sistemas operativos	Ingeniería en sistemas	123456789C	
26032	Auditoria informática	Ingeniería en sistemas	123456789C	

Nota. Interfaz abstracta para crear editar y eliminar nuevas asignaturas. Elaborado por: los autores.

Interfaz donde solo el administrador tendrá acceso para agregar un coordinador de asignatura, el cual más adelante podrá generar las prácticas de dicha asignatura. Adicionalmente el administrador podrá agregar docentes que pertenezcan a la asignatura sin embargo esta función también puede ser realizada por el usuario coordinador de asignatura. Ver figura 23.

Figura 23

Interfaz abstracta: creación de docente y asignación de coordinador de asignatura

The screenshot shows a web browser window with the URL `http://localhost:8080`. The page has a header with 'Inicio' and navigation links '< Atras' and 'Perfil'. The main content area is titled 'Asignatura' and contains two tabs: 'Docentes' and 'Temas'. The 'Docentes' tab is active, showing two sections: 'Coordinador' and 'Docente'. The 'Coordinador' section has a dropdown menu 'Seleccione coordinador' and a table with columns 'Cedula', 'Nombre', and 'Apellido'. The 'Docente' section has a dropdown menu 'Seleccione Docente' and a table with columns 'Cedula', 'Nombre', and 'acciones'. The table for 'Coordinador' has one row with values '1234567890', 'Jefferson', and 'Oña'. The table for 'Docente' has two rows with values '12345678', 'Javier Cobo', and 'acciones'.

Nota. Interfaz abstracta para añadir docentes y coordinadores de asignatura. Elaborado por: los autores.

2.3.6.3. Interfaces abstractas del Coordinador de asignatura Los temas de la asignatura están asociados a un tema principal o padre, del cual se derivan los subtemas, a su vez cada tema está asociado a una materia.

Figura 22

Interfaz abstracta coordinador de asignatura: Agregar temas

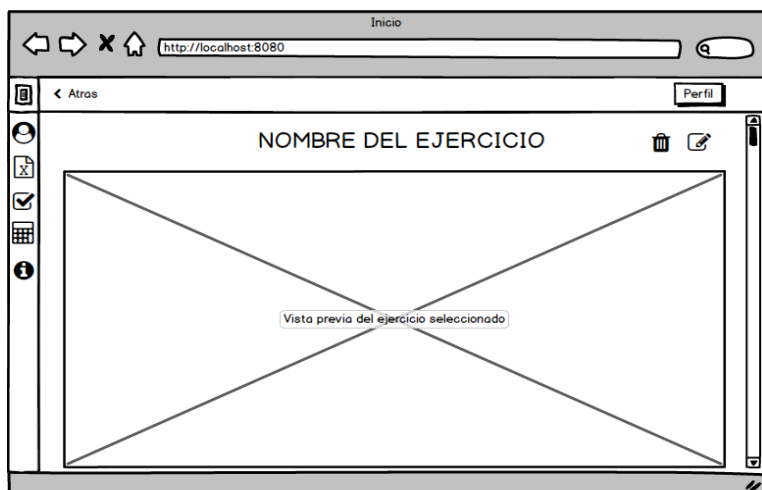
The screenshot shows a web browser window with the URL `http://localhost:8080`. The page has a header with 'Inicio' and navigation links '< Atras' and 'Perfil'. The main content area is titled 'Asignatura' and contains two tabs: 'Docentes' and 'Temas'. The 'Temas' tab is active, showing two sections: 'Nuevo tema' and 'Temas de la asignatura'. The 'Nuevo tema' section has a dropdown menu 'Seleccione tema padre' and a text input field 'Tema'. The 'Temas de la asignatura' section has a table with columns 'Base de datos' and 'acciones'. The table has three rows with values 'sql', 'nosql', and 'Programación'. The table for 'Nuevo tema' has one row with values 'sql', 'nosql', and 'Programación'.

Nota. Interfaz abstracta para agregar temas y sub temas. Elaborado por: los autores.

Los ejercicios se previsualizarán de acuerdo con el formato establecido por el claustro docente, de manera preliminar la pantalla mostrará el título, descripción, tema, ejercicios y solución.

Figura 24

Interfaz abstracta coordinador de asignatura: Mostrar ejercicio



Nota. Interfaz abstracta para tener una vista previa de un ejercicio en específico. Elaborado por: los autores.

Los ejercicios serán ingresados mediante un formulario el cual consta de un título, una descripción, un tema, el ejercicio en sí, un ejemplo y la solución propuesta por el docente para el ejercicio propuesto

Figura 25

Interfaz abstracta coordinador de asignatura: Crear ejercicios

Inicio

http://localhost:8080

< Atras Perfil

Gestionar Ejercicios

Nombre Asignatura Dificultad

Descripción

ejercicio

Ejemplo

Solución

Referencias

referencia	accic
Pressman R. S. & Troya J. M. (1988). In	<input type="checkbox"/>
Sommerville I. (2005). Ingeniería del s	<input type="checkbox"/>

Temas

Tema	acciones
Linux	<input type="checkbox"/>
Maquina virtual	<input type="checkbox"/>

Aceptar

Nota. Interfaz abstracta para agregar los datos necesarios de un nuevo ejercicio. Elaborado por: los autores.

La previsualización de las guías de laboratorio tendrá el formato establecido por el claustro docente en anteriores periodos académicos.

Figura 26

Interfaz abstracta coordinador de asignatura: Ver práctica



Nota. Interfaz abstracta para obtener una vista previa de la práctica. Elaborado por: los autores.

La creación de guías de laboratorio está centrada en un formulario donde se llenarán los datos informativos en campos de texto, con el fin de reusar las guías de laboratorio en cada periodo académico solo cambiando los ejercicios contenidos en las mismas. Ver figura 27.

Figura 27

Interfaz abstracta coordinador de asignatura: Crear ejercicios

The interface is titled "Práctica nueva" and is part of a web application running on http://localhost:8080. It features a sidebar with the following links: [Gestionar Usuarios](#), [Gestionar ejercicio](#), [Gestionar práctica](#), [Asignaturas](#), and [Reporte](#). The main form area contains the following elements:

- Formato:** A dropdown menu labeled "Seleccionar formato".
- Título:** A text input field.
- Objetivo:** A large text input field.
- Requisitos:** A text input field followed by a right-pointing arrow. To its right is a table:

requisito	acciones
Computador con sistema operativo	
Java JDK 1.8.	
- Instrucciones:** A text input field followed by a right-pointing arrow. To its right is a table:

instruccion	accio
Lea detenidamente cada uno de los e	
Plantee una solución a cada uno de l	
- Resultados obtenidos:** A text input field followed by a right-pointing arrow. To its right is a table:

resultado	accio
Los alumnos adquieren destreza para	
- Conclusiones:** A text input field followed by a right-pointing arrow. To its right is a table:

conclusiones	accio
Generar al menos una conclusión de l	
- Temas:** A dropdown menu labeled "Seleccionar temas". To its right is a table:

Temas de la practica	acciones
linux.	

At the bottom of the form is a large button labeled "Aceptar".

Nota. Interfaz abstracta para agregar los datos informativos de una nueva práctica.

Elaborado por: los autores.

El generar una guía es el paso final del aplicativo, por ende, una vez se tenga la plantilla de la guía de laboratorio creada, se procede a seleccionar los ejercicios que la conformaron, desplegados en un listado de acuerdo con la asignatura.

Figura 28

Interfaz abstracta coordinador de asignatura: Generar práctica

The screenshot shows a web application titled 'Generar Práctica'. It features a sidebar on the left with links: 'Gestionar Usuarios', 'Gestionar ejercicio', 'Gestionar práctica', 'Asignaturas', and 'Reporte'. The main content area has a header with 'Inicio', navigation buttons ('Atras', 'Perfil'), and a search bar. Below the header, there's a section for 'Práctica' with a dropdown menu 'Seleccionar práctica', a text input 'numero ejercicios', and a 'Crear nueva práctica' button. A table lists exercises with columns: 'Nombre', 'Dificultad', 'Elaborado por', 'veces usado', 'Descripcion', and 'acciones'. The table contains five rows of exercises related to 'Archivos en linux'. Below the table are buttons for 'Ver prácticas anteriores', 'Generar nuevamente', and 'Aceptar'. At the bottom, there's a large box labeled 'Vista previa del ejercicio seleccionado' which is currently empty.

Nombre	Dificultad	Elaborado por	veces usado	Descripcion	acciones
Archivos en linux	Baja	Javier cobo	1	manejor de archivos	
Archivos en linux	Baja	Javier cobo	2	manejor de archivos	
Archivos en linux	Media	Javier cobo	1	manejor de archivos	
Archivos en linux	Media	Javier cobo	1	manejor de archivos	
Archivos en linux	Alta	Javier cobo	2	manejor de archivos	

Nota. Interfaz abstracta para especificar el número de ejercicios que se desea en la práctica a generar. A su vez el coordinador podrá elegir o descartar los ejercicios como vea conveniente.

Elaborado por: los autores.

CAPÍTULO III

CONSTRUCCIÓN Y PRUEBAS

Este capítulo contiene las herramientas usadas, la construcción del aplicativo basado en la metodología SCRUM, se describe código y actividades realizadas en cada iteración del desarrollo a lo largo de 3 meses.

4.1 Herramientas

4.1.1 GitHub

“GitHub es un servicio comercial de alojamiento de repositorios Git remotos creado en el año 2008.” (López et. al, 2015). la principal función dentro del desarrollo fue el trabajo en equipo, debido a que la codificación realizada por cada miembro desarrollar es unificada en el repositorio compartido del proyecto, dando como resultado una constante actualización del aplicativo.

4.1.2 Visual Studio Code

Visual Studio Code es un editor de código fuente, permite personalización de acuerdo con las necesidades del proyecto o del desarrollador, adicionalmente es gratuito y de código abierto (Visual Studio Code, 2021).

4.1.3 PostMan

Postman es una aplicación la cual tiene como fin comprobar el funcionamiento de las APIs usadas dentro del aplicativo, ya que permite crear peticiones de manera rápida y fácil, adicionalmente ofrece un entorno de gestión colaborativo el cual encaja con el enfoque de desarrollo del presente proyecto (Postman, 2021).

4.1.4 MongoDB Compass

MongoDB Compass es un entorno gráfico, el cual permite visualizar y explorar los datos de manera intuitiva, adicionalmente permite modificar los datos (insertar, eliminar o modificar) (MongoDB, 2021).

4.2 Product Backlog

A fin de obtener los requerimientos, el equipo de desarrollo en conjunto con el coordinador del claustro docente levanto una lista de requerimientos, los cuales fueron plasmados en el Product backlog ver Tabla 20. Dicha tabla consta de 20 actividades a realizar.

Dicha tabla está constituida por.

- Historia/Tarea: nombre de la actividad o historia de usuario relacionada con la actividad a desarrollar.
- Tema: tema al cual está relacionado la actividad a desarrollar.
- Estado: estado de la actividad puede estar completado, pendiente o en desarrollo.
- Tiempo: tiempo estimado de desarrollo en días.
- Fecha de inicio: fecha de inicio de la actividad.
- Prioridad: prioridad de la actividad según el criterio del equipo de desarrollo.

Tabla 20

Product Backlog

Product Backlog						
ID	Historia/Tarea	Tema	Estado	Tiempo estimado (días)	Fecha inicio	Prioridad

I-01	Determinar herramientas de desarrollo	IDE	Completado	2	2020/12/21	Media
I-02	Instalación y configuración de IDE de desarrollo	IDE	Completado	3	2020/12/23	Alta
A-01	Estudiar distintas arquitecturas de software	Arquitectura de software	Completado	2	2020/12/28	Media
A-02	Establecer arquitectura de software para el aplicativo	Arquitectura de software	Completado	1	2020/12/30	Alta
B-01	Creación de base de datos	Base de datos	Completado	1	2020/01/31	Alta
R-01	HU-2 Tipos de usuarios	Registro usuarios	Completado	1	2021/01/05	Media
R-02	HU-3 Registro de usuarios	Registro usuarios	Completado	2	2021/01/06	Media
L-01	HU-4 Acceso de usuarios	Inicio de sesión	Completado	2	2021/01/08	Alta
D-01	HU-1 Imagen del sistema	Diseño visual	Completado	1	2021/01/11	Baja
MA-01	HU-8 Creación de periodos	Módulo administración	Completado	2	2021/01/18	Alta
MA-02	HU-9 Gestión de carreras	Módulo administración	Completado	2	2021/01/20	Baja
MA-03	HU-10 Designar coordinador de asignatura	Módulo administración	Completado	2	2021/01/22	Baja
As-01	HU-11 Gestión de asignatura	Asignatura	Completado	3	2021/01/25	Medio
AT-01	HU-12 Asignar temas a la asignatura	Asignar temas	Completado	3	2021/01/28	Baja
DD-01	HU-13 Designar docentes a la asignatura	Designar docentes	Completado	2	2021/02/01	Medio
RB-01	HU-14 Referencias bibliográficas	Referencias	Completado	2	2021/02/04	Baja
CE-01	HU-15 Crear ejercicio	Crear ejercicio	Completado	6	2021/02/08	Alta
GL-01	HU-1 Crear guía de laboratorio	Guía de laboratorio	Completado	6	2021/02/15	Alta
GL-02	HU-1 Generación de guía de laboratorio	Guía de laboratorio	Completado	6	2021/02/22	Alta
RE-01	HU-1 Generar reporte	Reporte	Completado	6	2021/03/01	Medio

Nota. Lista de requerimientos y tareas a realizar. Elaborado por: los autores.

4.3 Sprints

Cada sprint consistió en una semana de trabajo, las cuales fueron plasmadas de manera visual en un calendario. Al finalizar cada semana de trabajo, se realizó una reunión de revisión con el tutor, con el objetivo de evaluar el entregable correspondiente a esa semana y verificar si se necesitan aplicar cambios o si el tutor está conforme con lo entregado.

Las tareas realizadas en cada semana fueron plasmadas en una tabla donde están registradas las tareas completadas, los responsables y el tiempo de esfuerzo en horas por cada tarea, divididas en los días que duro el sprint.

En el mes de diciembre se llevaron a cabo 2 sprints, los cuales se pueden evidenciar en el anexo adjunto (Calendario sprints).

4.4 Sprint 1

En el primer sprint se tuvo como entregable el Ambiente de desarrollo, se determinó las herramientas a usar y la configuración del mismo, en este punto se estableció como editor de código Visual Code Studio, como aplicación de desarrollo de API se seleccionó PostMan y como aplicación gestora de base de datos MongoDB Compass.

Tabla 21

Tareas de sprint 1

Días					21/12	22/12	23/12	24/12	25/12
Horas trabajo pendiente					20	20	8	8	8
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo				
I-01	Determinar herramientas de desarrollo	IDE	Completado	Cobo Javier, Oña Jefferson	12	12			
I-02	Instalar Visual Code, Mongo DB Compass y Postman	IDE	Completado	Cobo Javier, Oña Jefferson	4	4	4		
I-02	Configurar Visual Code, Mongo DB Compass y Postman	IDE	Completado	Cobo Javier, Oña Jefferson	4	4	4	8	8

Nota. Tareas realizadas en el sprint 1. Elaborado por: los autores

4.4.1 Preparación del entorno

Previo al desarrollo se procedió a realizar la instalación de las herramientas de desarrollo a usar, siendo estas: Visual Code Studio, Postman y MongoDB Compass actuando como editor de código, gestor de APIs y entorno visual de base de datos respectivamente.

4.4.1.1. Entorno de MongoDB La gestión y creación de la base de datos se realizó en el gestor en la nube de base de datos MongoDB Atlas, a fin de mantener un entorno controlable, seguro y escalable adecuado al proyecto. MongoDB Atlas permite la creación de un clúster donde se alojará la base de datos, la opción gratuita M0 que brinda 512Mb de almacenamiento y procesamiento de CPU compartido fue la más adecuado debido a que el aplicativo no prevé el almacenamiento de una gran cantidad de datos, ya que son solo archivos de texto.

4.4.1.2. Entorno de NodeJS La instalación de NodeJS se realizó mediante el instalador de Windows en la versión 14.17 LTS, el resto de las configuraciones y creación se realiza mediante comandos por medio de la terminal de Visual Code Studio, la inclusión de librerías se realiza mediante el administrador de paquetes npm. Como resultado de ello obtenemos el archivo “server.js” en el cual se realiza la confirmación y se asignan las dependencias a usar dentro del aplicativo.

4.5 Sprint 2

Como recomendación y experiencia del tutor se inició con una arquitectura MVC con el uso de NodeJS pero por la dimensionalidad del aplicativo y debido a la necesidad del aplicativo de ser lo más mantenible posible a lo largo del tiempo, se decidió separar la lógica de negocio y el manejo de la base de datos en un programa, las interfaces de usuario en otro programa y mediante los protocolos REST comunicar ambos programas los cuales son denominados servicios de backend para la lógica de negocio y frontend para las interfaces.

Tabla 22*Tareas de sprint 2*

Días					28/12	29/12	30/12
Horas trabajo pendiente					16	16	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo		
A-01	Estudiar distintas arquitecturas de aplicaciones web	Arquitectura de software	Completado	Oña Jefferson	12	12	
A-02	Escoger la mejor arquitectura de software para el aplicativo	Arquitectura de software	Completado	Cobo Javier, Oña Jefferson	4	4	12

Nota. Tareas realizadas en el sprint 2. Elaborado por: los autores

En el mes de enero se llevaron a cabo 2 sprints, los cuales se pueden evidenciar en el anexo adjunto (Calendario sprints).

4.6 Sprint 3

El sprint tuvo como entregable el Registro de usuarios y su inicio de sesión en el aplicativo.

Tabla 23

Tareas de sprint 3

Días					31/12	04/01	05/01	06/01	07/01	08/01
Horas trabajo pendiente					17	13	19	17	15	13
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
B-01	Creación de base de datos en Mongo Atlas	Base de datos	Completado	Oña Jefferson	4	1	1	1	1	1
R-01	HU-2 Tipos de usuarios	Registro usuarios	Completado	Oña Jefferson, Cobo Javier	1	4				
R-02	HU-3 Registro de usuarios	Registro usuarios	Completado	Oña Jefferson, Cobo Javier	2	2	12			
R-02	Pruebas de registro de usuarios	Registro usuarios	Completado	Oña Jefferson, Cobo Javier	2	2	2	12		
L-01	HU-4 Acceso de usuarios	Inicio de sesión	Completado	Oña Jefferson, Cobo Javier	2	2	2	2	12	
L-01	Pruebas de inicio de sesión	Inicio de sesión	Completado	Oña Jefferson, Cobo Javier	2	2	2	2	2	12

Nota. Tareas realizadas en el sprint 3. Elaborado por: los autores.

4.6.1 Conexión a la base en MongoDB Atlas

La conexión a la base de datos de MongoDB se realiza mediante la función `connect`, se usó una variable de entorno “`process.env.DB_MONGO`”, con el fin de mantener las credenciales de acceso de manera privada.

Figura 29

Código conexión a base de datos

```
await mongoose.connect(process.env.DB_MONGO, {
  useNewUrlParser: true,
  useCreateIndex: true,
  useUnifiedTopology: true,
  useFindAndModify: false,
})
```

Nota. Fragmento de código de conexión a la base de datos del aplicativo. Elaborado por: los autores [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa/issues).

4.6.2 Encriptación de contraseña

Los datos de contraseña ingresados por el usuario se recogen mediante el código “`req.body`” y posteriormente son encriptados con la librería “`bcrypt`”, mediante la función `hash`.

Figura 30

Código encriptado de clave usuario

```
const usuario = new Usuario(req.body)
const salt = await bcryptjs.genSalt(10)
usuario.contrasena = await bcryptjs.hash(contrasena, salt)
```

Nota. Fragmento de código donde se encripta la contraseña del usuario. Elaborado por: los autores [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa/issues).

4.6.3 Verificación de seguridad mediante token

Cuando se realiza un inicio de sesión, si las credenciales de acceso son correctas, se procede a firmar el token con la función “jwt.sign”, el cual será enviado al frontend con el objetivo de verificar el inicio de sesión en cada petición que realice el usuario, a fin de verificar la seguridad en cada ejecución que realice el usuario.

Figura 31

Código de inicio de sesión mediante token

```
jwt.sign(payload, process.env.TOKEN,{ expiresIn: 36000 },(error, token) => {  
  if (error) throw error  
  res.status(200).json({ msg: 'login exitoso', token })  
})
```

Nota. Fragmento de código donde se puede evidenciar el inicio de sesión se realiza mediante un token. Elaborado por: los autores. [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa)

4.6.4 Entregables de sprint 3

Para la página de inicio, se mantuvo un diseño minimalista, donde se permita el ingreso de credenciales de acceso, siendo estas usuario y contraseña.

Figura 32

Pantalla de login

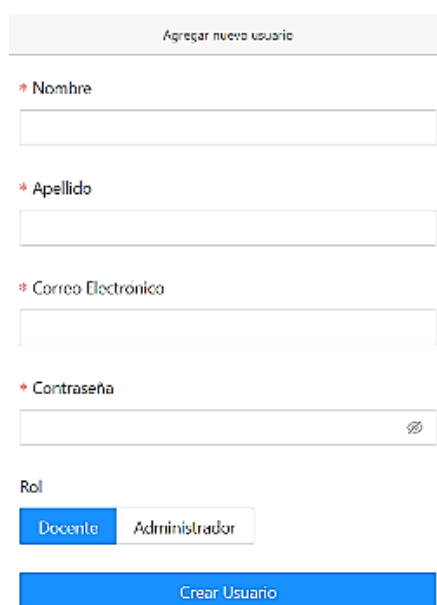


Nota. Pantalla de inicio en el aplicativo. Elaborado por: los autores.

La creación de usuarios solo puede ser realizada por el usuario administrador del sistema, mediante un formulario de los datos básicos del docente.

Figura 33

Registro de usuario



Formulario de registro de usuario. El formulario está encabezado por un botón gris que dice "Agregar nuevo usuario". A continuación, hay cuatro campos de texto con etiquetas precedidas por un asterisco rojo (*): "Nombre", "Apellido", "Correo Electrónico" y "Contraseña". El campo de "Contraseña" tiene un ícono de ojo para alternar la visibilidad. Debajo de estos campos, hay una sección titulada "Rol" con dos botones: "Docente" (en azul) y "Administrador" (en gris). Al final del formulario, hay un botón azul que dice "Crear Usuario".

Nota. Pantalla de registro en el aplicativo. Elaborado por: los autores.

A fin de validar el trabajo ya hecho y el código al completo, se puede remitir al repositorio de GitHub, donde se puede observar lo que se hizo al detalle.

<https://github.com/jeffqev/herramienta-colaborativa/issues?q=is%3Aissue+is%3Aclosed>

4.7 Sprint 4

El sprint tuvo como entregable el maquetado y la comunicación entre el backend y el frontend, adicionalmente la documentación generada en el aplicativo de cómo funcionan cada una de las API usadas.

Tabla 24

Tareas de sprint 4

Días					15/01	18/01	19/01	20/01	21/01	22/01
Horas trabajo pendiente					10	10	15	7	5	5
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
D-01	HU-1 Imagen del sistema	Maquetado	Completado	Cobo Javier	8	1	1	1	1	1
	Maquetado Minimalista	Maquetado	Completado	Cobo Javier	1	8	2	2		
	Documentación API		Completado	Oña Jefferson	1	1	12	4	4	4

Nota. Tareas realizadas en el sprint 4. Elaborado por: los autores.

4.7.1 Comunicación entre frontend y backend

La función de la figura 34, representa el estándar de comunicación entre el frontend y el backend, realizando peticiones de tipo get, post, put, delete y patch, dichas peticiones se realizan mediante la librería Axios, esta librería retorna una respuesta desde el backend mediante la función “dispatch”, en caso de ser satisfactoria la respuesta la información es enviada a la interfaz, caso contrario se envía un mensaje de error.

Figura 34

Comunicación entre interfaces

```
const iniciarSesion = async (datos) => {
  try {
    const respuesta = await clienteAxios.post(LOGIN, datos);
    dispatch({ type: LOGIN_EXITOSO, payload: respuesta?.data });

    // Obtener el usuario
    usuarioAutenticado();
  } catch (error) {
    const alerta = { msg: errorMsg(error), categoria: "danger" };
    dispatch({ type: LOGIN_ERROR, payload: alerta });
  }
};
```

Nota. Fragmento de código donde se realiza el intercambio de información entre backend y frontend. Elaborado por: los autores [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa).

4.7.2 Comunicación entre frontend y backend

Con el uso de la librería de Swagger, se creó una ruta donde se documentará el uso de las APIs, con un formato estandarizado ver Figura 35.

Figura 35

Documentación API

```
// configuracion de swagger
const swaggerSpecs = swaggerJsdoc(swaggerOptions)
app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpecs))
```

Nota. Fragmento de código donde se documenta la función de cada API. Elaborado por: los autores [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa).

La estructura de la documentación de un API está conformada por un título, una descripción y la ruta como información general, adicionalmente cuenta con los parámetros que puede recibir y la respuesta que puede enviar. En el caso de la figura 54, se puede evidenciar que puede recibir un correo de tipo String y una contraseña de tipo String y los códigos de respuesta,

Figura 36

Estructura documento APIs

```
@swagger
tags:
  name: Login
  description: Endpoint para inicio de sesion
/api/login:
  post:
    tags: [Login]
    requestBody:
      required: true
      content:
        application/json:
          schema:
            type: object
            properties:
              correo:
                type: string
                default: jeff@gmail.com
              contrasena:
                type: string
    responses:
      200:
        description: login exitoso
      400:
        description: usuario no existe / contraseña incorrecta / contraseña, correo es obligatorio
      500:
        description: hubo un error en el servidor
```

Nota. Contenido de la documentación de un APIs. Elaborado por: los autores

[Issues · jeffgev/herramienta-colaborativa \(github.com\)](https://github.com/jeffgev/herramienta-colaborativa).

La documentación de cada API está plasmada, junto con sus códigos de respuestas y su estructura, el acceso a los mismos se realiza mediante el aplicativo en: “<https://quicklab-api.grupoia.ec/api-docs/>”

Figura 37

Documentación de APIs

The screenshot displays the API documentation interface. It lists several endpoints categorized by function:

- Login** (Endpoint para inicio de sesion): POST /api/login
- Asignaturas** (Manejo de las asignaturas en el sistema): No specific endpoints listed.
- Calificaciones** (Manejo de las calificaciones de un ejercicio en el sistema):
 - POST /api/calificacion
 - GET /api/calificacion/ejercicio/{id}
 - GET /api/calificacion/{ejercicio}
 - PUT /api/calificacion/{id}
 - DELETE /api/calificacion/{id}
- Carreras** (Manejo de las carreras en el sistema):
 - POST /api/carrera
 - GET /api/carrera

An example JSON response is shown:

```
{
  "código": "1010",
  "nombre": "Sistemas operativos",
  "carrera": "60163b1352b5961950dc2720",
  "coordinador": "60163b1352b5961950dc2720",
  "docentes": [
    "60163b1352b5961950dc2720",
    "60163b1352b5961950dc2720",
    "60163b1352b5961950dc2720"
  ]
}
```

A table of responses is also provided:

Code	Description
201	Asignatura ingresada con éxito
400	Datos requeridos / La asignatura con ese codigo ya existe / Error al insertar en la base de datos
401	Permisos insuficientes para realizar la acción
500	hubo un error en el servidor

Nota. Documentación de APIs con su explicación. Elaborado por: los autores.

En el mes de febrero se llevaron a cabo 2 sprints, los cuales se pueden evidenciar en el anexo adjunto (Calendario sprints).

4.8 Sprint 5

El sprint tuvo como entregable las pantallas de creación, eliminación, actualización y lectura de periodos, carreras, denominado CRUD de operaciones.

Tabla 25

Tareas de sprint 5

					29/01	1/02	2/02	3/02	4/02	5/02
Horas trabajo pendiente					16	16	16	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
MA-01	HU-8 Creación de periodos	Módulo administración	Completado	Oña Jefferson	12	12	12			
MA-02	HU-9 Gestión de carreras	Módulo administración	Completado	Oña Jefferson	2	2	2	12	12	12

Nota. Tareas realizadas en el sprint 5. laborado por: los autores.

4.8.1 Operación insertar

A partir de este punto todas las clases encargadas de realizar las acciones en el backend por medio de las API, tendrán el mismo formato de codificación, las cuales recibirán información del frontend y procesa una acción de insertar, ver, eliminar o modificar estas operaciones son conocidas como CRUD.

Los datos enviados por el usuario son almacenados en la variable “datosCarrera”, para posteriormente ser almacenados en la base de datos mediante la función “save” de la librería Mongoose, enviando una respuesta al usuario de si fue exitoso o no la operación.

Figura 38

Operación Insertar

```
const datosCarrera = req.body
const carreraModel = new Carrera(datosCarrera)

// Guardar en la base de datos
await carreraModel.save((err, room) => {
  if (err) {
    res.status(400).send({ msg: 'Error al insertar en la base de datos' })
    return
  }
  res.status(201).json({ msg: 'carrera ingresada con exito', id: room.id })
})
} catch (error) {
  res.status(500).json({ msg: 'hubo un error en el servidor' })
}
```

Nota. Fragmento de código donde se inserta la información. Elaborado por: los autores

[Issues · jeffqev/herramienta-colaborativa \(github.com\)](#).

4.8.2 Operación buscar

Se almacena en la variable “carreras”, el resultado de la búsqueda realizada por la función “find” de Mongoose, retorno el resultado o en su defecto un mensaje de error.

Figura 39

Operación buscar

```
// buscar en la db
const carreras = await Carrera.find()
// si no hay datos retornar 404 not found
if (!carreras) {
  res.status(404).json({ msg: 'No se encontraron carreras' })
  return
}
// caso contrario retornar la lista
res.status(200).json({
  msg: 'Busqueda realizada con exito',
  data: carreras,
})
} catch (error) {
  res.status(500).json({ msg: 'hubo un error en el servidor' })
}
```

Nota. Fragmento de código donde se busca la información. Elaborado por: los autores

[Issues · jeffqev/herramienta-colaborativa \(github.com\)](#).

4.8.3 Operación actualizar

A fin de actualizar un registro, primero se verifica si existe en la base mediante la función “findById”, si es así se actualiza el registro con los nuevos datos mediante la función “findByIdAndUpdate”, o en su defecto se enviará un mensaje de error.

Figura 40

Operación actualizar

```
let carreraEncontrada = await Carrera.findById(req.params.id)
if (!carreraEncontrada) {
  res.status(404).json({ msg: 'Carrera ha modificar no encontrada' })
  return
}

// Modificar en la db
carreraEncontrada = await Carrera.findByIdAndUpdate(
  { _id: req.params.id },
  { $set: { carrera },
    { new: true }
  }
)
res
  .status(200)
  .json({ msg: 'carrera modificada con exito', data: carreraEncontrada })
} catch (error) {
  res.status(500).json({ msg: 'hubo un error en el servidor' })
}
```

Nota. Fragmento de código donde se actualiza la información. Elaborado por: los autores [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa).

4.8.4 Operación eliminar

Para eliminar un registro se busca en la base mediante la función “findById”, si el registro no está usado en otra tabla de la base, se ejecuta la función “findOneAndRemove” y el registro es eliminado, caso contrario envía una respuesta de error.

Figura 41

Operación eliminar

```
const carreraEncontrada = await Carrera.findById(req.params.id)
if (!carreraEncontrada) {
  res.status(404).json({ msg: 'Carrera a eliminar no encontrada' })
  return
}
// Revisar si se esta usando la carrera
const TieneAsignatura = await Asignatura.find({ carrera: req.params.id })
if (TieneAsignatura.length !== 0) {
  res
    .status(404)
    .json({ msg: 'La carrera cuenta con asignaturas asignadas' })
  return
}
// Eliminar en la db
await Carrera.findOneAndRemove({ _id: req.params.id })
res.status(200).json({ msg: 'Carrera eliminada con exito' })
} catch (error) {
  res.status(500).json({ msg: 'hubo un error en el servidor' })
}
```

Nota. Fragmento de código donde se elimina la información. Elaborado por: los autores

[Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa).

La creación de los periodos se establece con un nombre y rango de fecha, en la que durará el periodo, solo puede estar activo un periodo académico al mismo tiempo.

Figura 42

Creación de periodo

Agregar nuevo periodo

* Periodo:

* Fecha:

Inicio periodo

→

Fin periodo

Guardar Periodo

El listado de periodos se puede visualizar donde solo un periodo puede estar activo al mismo tiempo.

Figura 43

Pantalla de activación de periodo

Periodo	Inicio periodo	Fin periodo	Estado
60	15/05/2021	26/06/2021	ACTIVAR  
59	01/04/2021	30/05/2021	ACTIVO
58	01/03/2021	28/08/2022	ACTIVAR  
57	01/09/2020	28/02/2021	ACTIVAR  
56	01/04/2021	31/05/2021	ACTIVAR  

Nota. Periodos creados en el aplicativo y activación de uno solo. Elaborado por: los autores.

La creación de una carrera nueva se realiza de manera simple, solo con el nombre, es este proyecto por su alcance solo se tiene las carreras de sistemas y computación, pero tiene la posibilidad de ser extrapolado a otras carreras.

Figura 44

Pantalla de creación de carreras

Agregar Nueva Carrera

Carrera

Nombre

Agregar

Nota. Formulario de creación de carreras. Elaborado por: los autores.

4.9 Sprint 6

El sprint tuvo como entregable agregar asignaturas, adicionalmente solo los usuarios con permisos de administrador pueden modificar los registros de carreras dentro del aplicativo.

Tabla 26

Tareas de sprint 6

Días					12/02	15/02	16/02	17/02	18/02	19/02
Horas trabajo pendiente					14	14	4	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
As-01	HU-11 Gestión de asignatura	Módulo asignaturas	Completado	Oña Jefferson	12	12	2			
MA-03	HU-10 Designar coordinador	Módulo administración	Completado	Cobo Javier	2	2	2	2	12	12

Nota. Tareas realizadas en el sprint 6. Elaborado por: los autores.

4.9.1 Validar permisos

Esto es un middleware, donde se toma el rol del usuario mediante el token, si dicho usuario tiene el rol correspondiente a la función, se le da acceso, caso contrario se envía un mensaje de error.

Figura 45

Validar usuario administrador

```
module.exports = (req, res, next) => {  
  ...  
  const { rol } = req.logueado  
  
  if (rol !== 'administrador') {  
    res  
      .status(401)  
      .json({ msg: 'Permisos insuficientes para realizar la accion' })  
    return  
  }  
  next()  
}
```

Nota. Fragmento de código para validar usuario administrador. Elaborado por: los autores

[Issues · jeffqev/herramienta-colaborativa \(github.com\)](#).

En este punto se realiza la validación del middleware, en caso de tener el rol, se crea la asignatura caso contrario retorno un mensaje de error.

Figura 46

Creación de asignatura

```
router.post(  
  '/',  
  authToken,  
  administrador,  
  [check('codigo', 'El código es requerido').not().isEmpty()],  
  asignaturaController.crearAsignatura  
)  
  
router.get('/', authToken, asignaturaController.buscarAsignaturas)
```

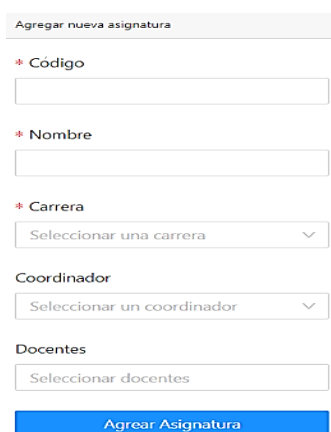
Nota. Fragmento de código para crear asignaturas. Elaborado por: los autores

[Issues · jeffqev/herramienta-colaborativa \(github.com\)](#).

La creación de asignaturas se realiza con el siguiente formulario, adicionalmente en el mismo formulario el coordinador de esta.

Figura 47

Pantalla creación asignatura



Formulario de creación de asignatura:

- Título: Agregar nueva asignatura
- * Código:
- * Nombre:
- * Carrera:
- Coordinador:
- Docentes:
- Botón: Agregar Asignatura

Nota. Formulario de creación de asignatura y asignación de docentes. Elaborado por: los autores.

En el mes de marzo se llevaron a cabo 3 sprints, los cuales se pueden evidenciar en la tabla 38.

4.10 Sprint 7

El sprint tuvo como entregable el CRUD de operaciones de crear referencias y designar docentes a la materia que imparten clases.

Tabla 27

Tareas de sprint 7

Días					26/02	01/03	02/03	03/03	04/03	05/03
Horas trabajo pendiente					14	14	14	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
DD-01	HU-13 Designar docentes a la asignatura	Designar docentes	Completado	Oña Jefferson	12	12	12			
RB-01	HU-14 Referencias bibliográficas	Referencias	Completado	Cobo Javier	2	2	2	12	12	12

Nota. Tareas realizadas en el sprint 7. Elaborado por: los autores.

4.10.1 Operación patch

Las modificaciones parciales en una clase se harán mediante el uso de la operación patch la cual no envió información, sino que asigna parcialmente datos que ya se encuentran en el aplicativo. La función patch permite agregar docentes a la asignatura siempre y cuando tenga los permisos adecuados.

Figura 48

Asignar docentes

```
router.put('/:id', authToken, administrador, asignaturaController.modificarAsignatura )

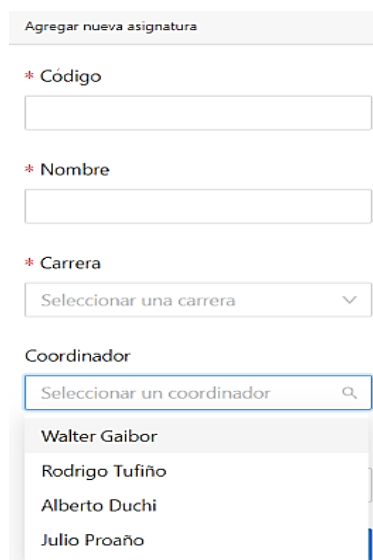
router.patch(
  '/docentes/:id',
  authToken,
  [check('docentes', 'Agregar al menos un docente').not().isEmpty()],
  asignaturaController.docentesAsignatura
```

Nota. Fragmento de código donde se puede agregar docentes a la asignatura. Elaborado por: los autores [Issues · jeffqev/herramienta-colaborativa \(github.com\)](https://github.com/jeffqev/herramienta-colaborativa).

La asignación del docente coordinador, se realiza al momento de crear la asignatura, con la posibilidad de poder ser cambiado.

Figura 49

Pantalla creación asignatura



Formulario de creación de asignatura:

- Título: Agregar nueva asignatura
- * Código:
- * Nombre:
- * Carrera:
- Coordinador: (con icono de lupa)
- Lista de coordinadores: Walter Gaibor, Rodrigo Tufiño, Alberto Duchi, Julio Proaño


Nota. Formulario de creación de asignatura y asignación de docente coordinador. Elaborado por: los autores.

El formato de ingreso de referencias es llenado en un formulario, donde se ingresarán los datos de la referencia y se concatenara automáticamente con el formato APA. Ver figura 50.

Figura 50

Pantalla de bibliografías

Referencias

	Título	Autores	Publicación	Tipo	Acciones
—	Sistemas digitales: principios, análisis y diseño	Héctor Flórez.	2001	Libro	 
Flórez, H. (2001). Sistemas digitales: principios, análisis y diseño					
—	Fundamentos de sistemas digitales	Medium.	2001	Web	 
Medium. (2001). Fundamentos de sistemas digitales. [online]					

Nota. Lista de referencias con su nombre y formato APA. Elaborado por: los autores.

Figura 51

Pantalla creación referencias

Agregar nueva referencia

Tipo: ☐ Libro ☐ Sitio Web

* Título

Sistemas digitales: principios, análisis

+ Agregar Autor

Año de publicación

Edición

Editorial

Agrear Referencia

Nota. Formulario de creación de citas bibliográficas. Elaborado por: los autores

4.11 Sprint 8

El sprint tuvo como entregable uno de los principales componentes del aplicativo, el cual es el módulo de creación de ejercicios.

Tabla 28

Tareas de sprint 8

Días					12/03	15/03	16/03	17/03	18/03	19/03
Horas trabajo pendiente					12	12	12	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
CE-01	HU-15 Crear ejercicio	Crear ejercicio	Completado	Oña Jefferson, Javier Cobo	12	12	12	12	12	12

Nota. Tareas realizadas en el sprint 8. Elaborado por: los autores.

4.11.1 Creador de ejercicios

El creador de ejercicios fue realizado con el componente gratuito “tinymce”, donde se agrupan todos los plugin que se usarán, adicionalmente se envía una serie de combinaciones como tamaño e idioma, siendo lo más importante el api key proveído por la página de tinymce para poder concatenar los datos informativos del ejercicio y el ejercicio en sí mediante editores de texto enriquecido. Ver figura 52.

Figura 52

Editor tinymce

```
import { Editor } from "@tinymce/tinymce-react";

function RichText({ htmlEditor, setHtmlEditor }) {
  const handleChange = (content, editor) => {
    setHtmlEditor(content);
  };
  return (
    <Editor
      value={htmlEditor}
      apiKey={process.env.REACT_APP_RICHTEXT_KEY}
      init={{
        height: 300,
        language: "es",
        plugins: [
          "advlist autolink lists link image charmap print preview anchor",
          "searchreplace visualblocks code fullscreen",
          "insertdatetime table paste code wordcount",
        ],
        toolbar: `fullscreen | undo redo | formatselect | bold italic backcolor |
          alignleft aligncenter alignright alignjustify |
          bullist numlist outdent indent | removeformat | charmap |
          table | image | searchreplace`,
        images_upload_handler: example_image_upload_handler,
      }}
      onEditorChange={handleChange}
    />
  );
}
```

Nota. Fragmento de código donde el editor de ejercicios es ejecutado. Elaborado por: los autores [Realizar el módulo de ejercicios · Issue #6 · jeffqev/herramienta-colaborativa](#)

4.11.2 Subir imágenes al ejercicio

Al ser un editor de texto enriquecido, permite agregar imágenes, por lo tanto, se crea una ruta donde se guardarán las imágenes en el servidor mediante la variable “uploadPath” con un nombre único, el código retorna la dirección de la imagen con el nuevo nombre generado. Ver figura 53.

Figura 53

Carga de imágenes

```
app.post('/upload', (req, res) => {
  if (!req.files || Object.keys(req.files).length === 0) {
    res.status(400).send('No se ha enviado un archivo.')
    return
  }
  const { file } = req.files
  const extension = file.name.split('.')
  const nuevoNombre = `${file.md5}.${extension[extension.length - 1]}`
  const uploadPath = `${__dirname}/uploads/${nuevoNombre}`

  file.mv(uploadPath, (err) => {
    if (err) {
      res.status(500).send(err)
    }
  })

  res.json({ location: `${req.protocol}://${req.get('host')}/images/${nuevoNombre}` })
})
```

Nota. Fragmento de código de carga de imágenes desde el editor de texto. Elaborado por: los autores [Realizar el modulo de ejercicios · Issue #6 · jeffqev/herramienta-colaborativa](#)

Los ejercicios en dentro del claustro docente, se plasman como se puede ver Figura 54

Figura 54

Formato de ejercicios

ACTIVIDADES A DESARROLLAR:

Yadira

1. Figura Geométrica Cuadrada

Representar el siguiente algoritmo en pseudocódigo y mediante un diagrama de flujo de datos:

Leer el lado de un cuadro, calcular y desplegar el perímetro y área del cuadrado

Solución:

Pseudocódigo

```
Algoritmo Cuadrado
  definir lado, area, perimetro Como Real
  Escribir "Ingrese el lado del cuadrado:"
  Leer lado
  perimetro ← lado * 4
  area ← lado * 2
  Escribir "RESULTADOS"
  Escribir "Perimetro:", perimetro
  Escribir "Area:", area
```

FinAlgoritmo

```
PSaint - Ejecutando proceso CUADRADO
*** Ejecución Iniciada. ***
Ingrese el lado del cuadrado:
> 4.5
RESULTADOS
Perimetro:18
Area:20.25
*** Ejecución Finalizada. ***
```

Diagrama de Flujo de Datos

Actividad para desarrollar descrita por el docente.

Ejercicio soportado con imágenes, código, tablas o fórmulas


Nota. Formato de ejercicios en guías de laboratorio. Fuente: Guías de laboratorio claustro docente de programación de la Universidad Politécnica Salesiana.

La creación de ejercicios está concentrada en un formulario, donde hay campos de texto enriquecido que permiten la inserción de archivos multimedia como imágenes, el editor de texto es similar a editores comunes en el mercado. Ver figura 55.

Todos los ejercicios creados se pueden visualizar por parte del usuario creador, a partir de este punto todas las tablas mantiene la estructura que se puede ver en la figura 56, donde se puede filtrar la información o realizar la acción de borrar, editar o ver.

Figura 56

Pantalla de listado de ejercicios

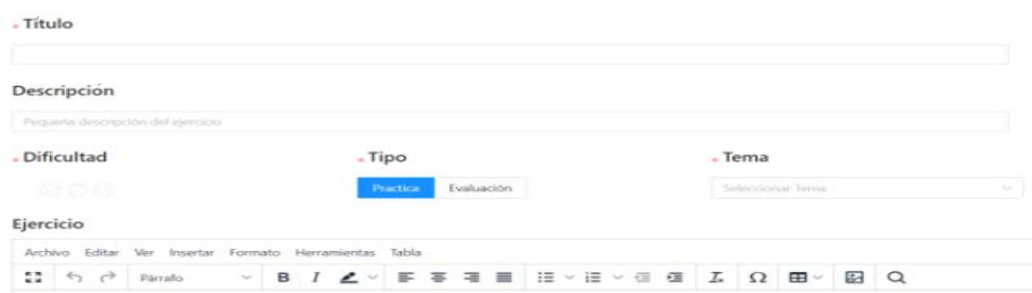


	Titulo	Tema	Dificultad	Calificación	Tipo	Archivado	Acciones
+	Aaaaaa	Sentencias de repeticion	Medio	★★★★★	Evaluación	Sin Archivar	
+	Año bisiesto	Algoritmo avanzado	Fácil	★★★★★	Practica	Sin Archivar	

Nota. Lista de ejercicios creados en el aplicativo. Elaborado por: los autores [Generar practicas by jeffqev · Pull Request #15 · jeffqev/herramienta-colaborativa \(github.com\)](#).

Figura 55

Pantalla de creación de ejercicios



Título

Descripción

Dificultad

Tipo

Tema

Ejercicio

Nota. Formulario de datos para creación de ejercicios. Elaborado por: los autores [Generar practicas by jeffqev · Pull Request #15 · jeffqev/herramienta-colaborativa \(github.com\)](#).

A fin de validar el trabajo ya hecho y el código al completo, se puede remitir al repositorio de GitHub, donde se puede observar lo que se hizo al detalle.

<https://github.com/jeffqev/herramienta-colaborativa/issues/6>

4.12 Sprint 9

El sprint tuvo como entregable el formulario donde se engrosarán los datos informativos de la guía de laboratorio como el título, objetivos, etc.

Tabla 29

Tareas de sprint 9

Días					22/03	23/03	24/03	25/03	26/03	29/03
Horas trabajo pendiente					12	12	12	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
GL-01	HU-1 Crear guía de laboratorio	Plantillas	Completado	Oña Jefferson, Javier Cobo	12	12	12	12	12	12

Nota. Tareas realizadas en el sprint 9. Elaborado por: los autores.

4.12.1 Creación guía del laboratorio

La creación de la guía de laboratorio se realiza mediante una operación CRUD de inserción mediante un formulario

Figura 57

Creación guía de laboratorio

```
return (  
  <div className="mt-3 ms-3">  
    <h3 className=" mt-3 text-center"> Nueva practica</h3>  
    <Form  
      // {...formItemLayoutWithoutLabel}  
      form={form}  
      name="practicassform"  
      onFinish={onFinish}  
      layout="vertical"  
      style={{ paddingLeft: "5%", paddingRight: "5%" }}  
    >  
      <Form.Item  
        name={["plantilla", "titulo"]}  
        label={<Title level={4}>Titulo</Title>}  
        rules={[  
          {  
            required: true,  
            message: "El título es obligatorio",  
          }  
        ]}  
      />  
    </Form>  
  </div>  
)
```

Nota. Formulario de creación de guías de laboratorio. Elaborado por: los autores [Generar practicas by jeffqev · Pull Request #15 · jeffqev/herramienta-colaborativa \(github.com\)](#)

La creación de una guía de laboratorio tiene como objetivo el completar los campos informativos de la misma, los cuales cambian en menor medida entre periodos académicos.

Figura 58

Pantalla de creación de guías de laboratorio

Nueva practica

• Título

• Formato • Número Practica • Tema

Seleccionar un formato Seleccionar tema

Objetivos

Objetivos

Agregar nuevo objetivo

Agregar nuevo objetivo

+ Agregar Objetivos

Nota. Formulario de creación de guías de laboratorio. Elaborado por: los autores [Generar practicas by jeffqev · Pull Request #15 · jeffqev/herramienta-colaborativa \(github.com\)](#).

A fin de validar el trabajo ya hecho y el código al completo, se puede remitir al repositorio de GitHub, donde se puede observar lo que se hizo al detalle.

<https://github.com/jeffqev/herramienta-colaborativa/pull/15>

En el mes de abril se llevaron a cabo 2 sprints, los cuales se pueden evidenciar en el anexo adjunto (Calendario sprints).

4.13 Sprint 10

El sprint tuvo como entregable en la generación de guías de laboratorio en base a los ejercicios que se hayan ingresado por materia, esta selección es manual, en base a la necesidad del docente.

Tabla 30

Tareas de sprint 10

Días					02/04	05/04	06/04	07/04	08/04	09/04
Horas trabajo pendiente					12	12	12	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
GL-02	HU-1 Generación de guía de laboratorio	Guía de laboratorio	Completado	Oña Jefferson, Javier Cobo	12	12	12	12	12	12

Nota. Tareas realizadas en el sprint 10. Elaborado por: los autores.

4.13.1 Creación de guía de laboratorio en pdf

Mediante la librería HTML-pdf se convierte el maquetado HTML de los datos informativos de la guía junto al HTML de los ejercicios a un archivo pdf que se enviará al cliente para su descarga. La función utilizada es “create”, la cual anexa todos los datos HTML y crea un archivo pdf.

Figura 59

Generación de guías de laboratorio

```
pdf.create(content, config).toStream((err, stream) => {
  if (err) {
    res.json({ msg: 'error' })
  }
  res.setHeader('Content-Type', 'application/pdf')
  res.setHeader('Content-Disposition', 'attachment; filename=data.pdf')
  stream.pipe(res)
})
```

Nota. Fragmento de código donde se concatenan los ejercicios en un solo formato de texto.

Elaborado por: los autores [Generar practicas by jeffqev · Pull Request #15 · jeffqev/herramienta-colaborativa \(github.com\)](#).

Generar una guía es el producto final del aplicativo, en este apartado se selecciona los ejercicios con los cuales la guía será completada, dichos ejercicios son seleccionados acorde el docente considere adecuado, el aplicativo ofrece parámetros de selección como dificultad, veces que ha sido usado el ejercicio e inclusive la calificación. Ver figura 60.

Figura 60

Pantalla de selección de ejercicios

<input checked="" type="checkbox"/>	Titulo	Descripcion	Tema	Dificultad	Utilizado	
<input checked="" type="checkbox"/>	Reglas del algebra booleana		Lógica de programación	Medio	1	

Nota. Pantalla de selección de ejercicios. Elaborado por: los autores.

A fin de validar el trabajo ya hecho y el código al completo, se puede remitir al repositorio de GitHub, donde se puede observar lo que se hizo al detalle <https://github.com/jeffqev/herramienta-colaborativa/pull/15>

4.14 Sprint 11

El sprint tuvo como entregable el reporte de uso de ejercicios en el periodo activo.

Tabla 31

Tareas de sprint 11

Días					16/04	19/04	20/04	21/04	22/04	23/04
Horas trabajo pendiente					12	12	12	12	12	12
Backlog ID	Historia/Tarea	Tema	Estado	Responsable	Esfuerzo					
RE-01	HU-1 Generar reporte	Reporte	Completado	Oña Jefferson, Javier Cobo	12	12	12	12	12	12

Nota. Tareas realizadas en el sprint 11. Elaborado por: los autores.

4.14.1 Generación de gráficas para reporte

Mediante el uso de la librería charts, se genera las gráficas a través de la información enviada por el backend, la cual realiza una consulta en la base, en base a los datos consultados se envía como parámetros para generar gráficas informativas.

Figura 61

Generación de gráficas

```
var config = {
  data: data,
  xField: "titulo",
  yField: "usado",
  label: {
    position: "middle",
    style: {
      fill: "#FFFFFF",
      opacity: 0.6,
    },
  },
  meta: {
    titulo: { alias: "Titulo" },
    usado: { alias: "Veces usado" },
  },
};

return <Column {...config} />;
```

Nota. Fragmento de código de reporte. Elaborado por: los autores [Reportes by jeffqev · Pull](#)

[Request #16 · jeffqev/herramienta-colaborativa \(github.com\)](#).

4.14.2 Generación de gráficas para reporte

El reporte de los ejercicios creados por cada docente se recopila por materia y es actualizado de manera constante.

Figura 62

Pantalla de reportes



Nota. Pantalla de reportes en el sistema. Elaborado por: los autores [Reportes by jeffqev · Pull Request #16 · jeffqev/herramienta-colaborativa \(github.com\)](#).

A fin de validar el trabajo ya hecho y el código al completo, se puede remitir al repositorio de GitHub, donde se puede observar lo que se hizo al detalle <https://github.com/jeffqev/herramienta-colaborativa/pull/16>

4.15 Pruebas

Las pruebas se realizaron con la finalidad de comprobar el funcionamiento del aplicativo web

4.15.1 Pruebas de caja negra

Las pruebas de caja negra se realizaron en los procesos de más importancia del aplicativo, a fin de buscar errores que afecten al usuario final, dichos errores fueron encontrados

y se les dio solución. Las tablas describen el problema identificado, la solución y el código de dicha solución. Ver Tabla 32.

Tabla 32

Prueba de caja negra 1

Prueba	
Número	1
Prioridad	Alta
Descripción	Eliminar una carrera, la carrera es eliminada aun cuando esta tiene asignadas materias
Archivo	CarreraController
Solución	Se realizo una búsqueda en todas las tablas para verificar que el registro a eliminar no se encuentra en uso.
Código	<pre> const TieneAsignatura = await Asignatura.find({ carrera: req.params.id }) if (TieneAsignatura.length !== 0) { res .status(404) .json({ msg: 'La carrera cuenta con asignaturas asignadas' }) return } // Eliminar en la db await Carrera.findOneAndRemove({ _id: req.params.id }) </pre>

Nota. Detalles de la prueba de la eliminación de una carrera. Elaborado por: los autores.

Tabla 33

Prueba de caja negra 2

Prueba	
Número	2
Prioridad	Alta
Descripción	Activar un periodo en 2 ventanas diferentes
Archivo	periodoController
Solución	Primero se actualizo todos los periodos como inactivos para a continuación activar el seleccionado.

Código	<pre>// Desactivar actual periodoEncontrado = await Periodo.updateMany({ estado: true }, { \$set: { estado: false } }, { new: true }) // Activar periodo por ID periodoEncontrado = await Periodo.findByIdAndUpdate({ _id: req.params.id }, { \$set: { estado: true } }, { new: true })</pre>
---------------	--

Nota. Detalles de la prueba activar un periodo académico al mismo tiempo. Elaborado por: los autores.

Tabla 34

Prueba de caja negra 3

Prueba	
Número	3
Prioridad	Alta
Descripción	Eliminar temas que cuentan con temas hijos
Archivo	temaController
Solución	Buscar si dicho tema cuenta con al menos un tema hijo relacionado
Código	<pre>// Buscar temas hijos por id const temaUtilizado = await Tema.find({ padre: req.params.id }) if (temaUtilizado.length !== 0) { res .status(400) .json({ msg: 'No se puede eliminar el tema cuenta con temas hijos ' }) return }</pre>

Nota. Detalles de la prueba de la eliminación de temas que cuentan con subtemas. Elaborado por: los autores.

Tabla 35

Prueba de caja negra 4

Prueba	
Número	4
Prioridad	Alta
Descripción	Modificar una asignatura como usuario coordinador
Archivo	asignaturaController
Solución	Se implemento un endpoint en el API para que un usuario coordinador solo pueda editar los usuarios que pertenecen a la asignatura y no otra información de la asignatura
Código	<pre> try { // Guardar los docentes enviados por request body const { docentes } = req.body const nuevosDatos = {} if (docentes) { nuevosDatos.docentes = docentes } // Revisar si existe por el id enviado let asignaturaEncontrada = await Asignatura.findById(req.params.id) if (!asignaturaEncontrada) { res.status(404).json({ msg: 'Asignatura a modificar no encontrada' }) return } // Modificar en la db asignaturaEncontrada = await Asignatura.findByIdAndUpdate({ _id: req.params.id }, { \$set: nuevosDatos }, { new: true }) res.status(200).json({ msg: 'Docentes actualizados con exito', data: asignaturaEncontrada, }) } catch (error) { res.status(500).json({ msg: 'hubo un error en el servidor' }) } </pre>

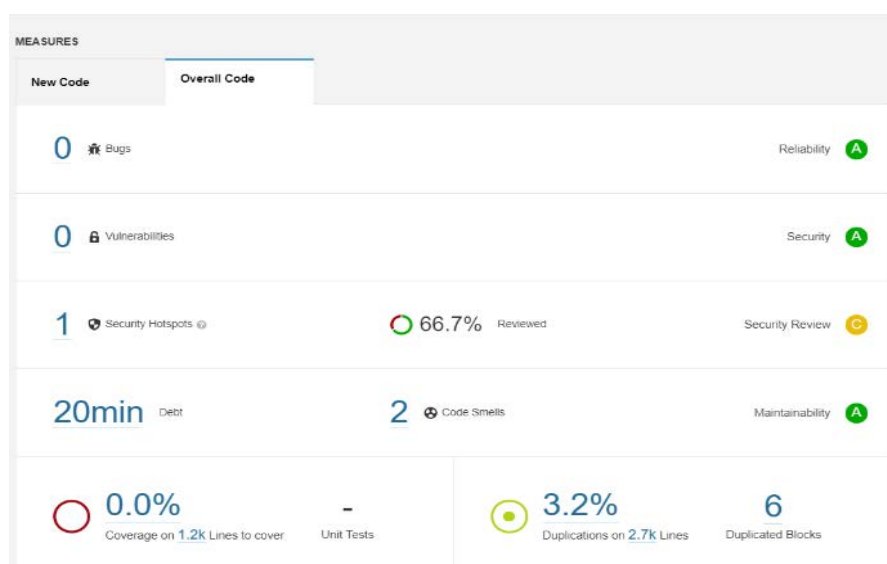
Nota. Detalles de la prueba comprobación de roles en acciones solo una acción. Elaborado por: los autores.

4.15.2 Pruebas del código

A fin de evaluar el código del aplicativo, se realizaron pruebas en SonarQube, el cual es una herramienta de análisis de código fuente basado en métricas. El resultado de las pruebas dio como resultado una advertencia de seguridad, dos posibles mejoras en código y líneas duplicadas. Ver figura 63.

Figura 63

Resumen de las pruebas de código fuente



Nota. resultado de las calificaciones asignadas por SonarQube. Fuente: SonarQube.

Se identificó un 3,2% de líneas de código duplicadas en todo el código fuente del aplicativo, el mayor porcentaje de duplicidad se encontró en los controladores de ejercicios y prácticas, debido a que su codificación contiene las partes más relevantes del aplicativo web. Ver figura 64.

Figura 64

Duplicidad de líneas de código

Duplicated Lines (%) 3.2%		
	Duplicated Lines (%)	Duplicated Lines
 server/controllers/ejercicioController.js	27.7%	98
 server/controllers/practicaController.js	19.0%	60
There are 54 hidden components with a score of 0.0%. Show Them		

Nota. resultado de la métrica de duplicidad de código arrojada por SonarQube. Fuente: SonarQube.

La advertencia de seguridad del backend, se debe a que la conexión a la API es de libre acceso a cualquier dominio que haga una petición, a pesar de que el riesgo es bajo, esta advertencia se pudo mitigar asignando un solo dominio el cual realizará las peticiones.

Figura 65


Resultado de prueba de seguridad

Make sure that enabling CORS is safe here.



[Add Comment](#) [Open in IDE](#) [Get Permalink](#)

Category **Insecure Configuration**

Review priority **LOW**

Assignee **Not assigned** 

Status: To review
This Security Hotspot needs to be reviewed to assess whether the code poses a risk

 server/server.js 

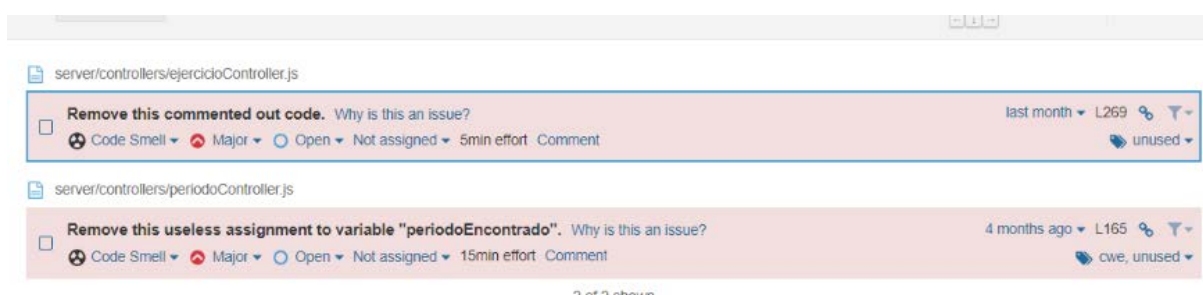
```
13
14 conectarDB()
15
16 // configuracion de express con cors
17 const app = express()
18 app.use(cors())
19 app.use(express.json({ extended: true }))
20
21 // configuracion de swagger
22 const swaggerSpecs = swaggerJsdoc(swaggerOptions)
23 app.use('/api-docs', swaggerUi.serve, swaggerUi.setup(swaggerSpecs))
```

Nota. resultado de la métrica de seguridad arrojada por SonarQube. Fuente: SonarQube.

Adicionalmente se identificó código comentando el cual no estaba siendo usado en la clase “ejercicioController” y se identificó una variable la cual no está siendo usada en la clase “periodoController”.

Figura 66

Código no usado



Nota. resultado de la métrica de código arrojada por SonarQube. Fuente: SonarQube.

4.15.3 Pruebas de estrés

Estimando el pequeño número de docentes pertenecientes al claustro docente de programación que harán uso del aplicativo, no habría problemas de rendimiento por concurrencia, pero como el aplicativo se encontrara alojado en un dominio de acceso público en internet, fue conveniente realizar pruebas de estrés, adicionalmente como se tiene pensado extender su uso a otras carreras se establecieron tres escenarios a fin de medir la concurrencia máxima de usuarios y el tiempo de ejecución. Las pruebas fueron realizadas en los procesos más importantes del aplicativo siendo estos la generación de una guía y la creación de ejercicios.

Se realizó el recorrido del proceso desde el inicio de sesión hasta la creación de guías de laboratorio y ejercicios, cada tres segundos se ingresó un nuevo usuario.

Escenario con usuarios 25

El tiempo promedio de los procesos con 25 usuarios fue de 514 milisegundos para la creación de un ejercicio y 2768 milisegundos para la generación de una guía de laboratorio con una concurrencia de 1 usuario cada 3 segundos. Dando como resultado un porcentaje de error del 0% siendo óptimo para el uso por parte del claustro docente de programación. Ver figura 67 y 68.

Figura 67

Reporte resumen proceso crear ejercicios

Etiqueta	# Muestras	Media	Min	Máx	% Error	Rendimiento
Login	25	450	179	675	0,00%	7,4/sec
materias coordinador	25	559	294	890	0,00%	6,7/sec
materias docente	25	611	385	1075	0,00%	6,5/sec
ejercicios	25	277	166	475	0,00%	6,9/sec
ejercicio	25	673	363	2770	0,00%	5,8/sec
Total	125	514	166	2770	0,00%	22,8/sec

Nota. Resumen del reporte generado por Jmeter. Elaborado por: los autores.

Figura 68

Reporte resumen proceso crear guía de laboratorio

Etiqueta	# Muestras	Media	Min	Máx	% Error	Rendimiento
Login	25	571	303	1276	0,00%	6,8/sec
materias coordinador	25	874	378	1323	0,00%	5,8/sec
materias docente	25	1012	373	1584	0,00%	5,0/sec
practica	25	1615	1024	2639	0,00%	3,9/sec
practica pdf	25	9767	2337	13276	0,00%	1,6/sec
Total	125	2768	303	13276	0,00%	7,1/sec

Nota. Resumen del reporte generado por Jmeter. Elaborado por: los autores.

Escenario con usuarios 100

El tiempo promedio de los procesos con 100 usuarios fue de 3504 milisegundos para la creación de un ejercicio y 13626 milisegundos para la generación de una guía de laboratorio con una concurrencia de 1 usuario cada 3 segundos. Dando como resultado un porcentaje de error del 18,60% en la generación de una guía, donde solo se descargaron 7 guías de laboratorio,

esto es lo esperable debido a que se está generando una guía cada 3 segundos lo cual no es lo esperable en el aplicativo.

Figura 69

Reporte resumen proceso crear ejercicios

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error
Login	100	3971	462	6904	1842,82	0,00%
materias coordinador	100	5561	1761	8362	1847,71	0,00%
materias docente	100	3138	2827	8078	894,35	0,00%
ejercicios	100	1452	912	1807	233,26	0,00%
ejercicio	100	3396	2640	4353	524,75	0,00%
Total	500	3504	462	8362	1829,81	0,00%

Nota. Resumen del reporte generado por Jmeter. Elaborado por: los autores.

Figura 70

Reporte resumen proceso crear guía de laboratorio

Etiqueta	# Muestras	Media	Mín	Máx	Desv. Estándar	% Error
Login	100	5215	493	9000	2536,47	0,00%
materias coordinador	100	6929	3399	10702	2327,14	0,00%
materias docente	100	3714	3290	9240	894,33	0,00%
practica	100	9070	5595	11246	1188,46	0,00%
practica pdf	100	43205	3664	49720	9620,13	93,00%
Total	500	13626	493	49720	15595,64	18,60%

Nota. Resumen del reporte generado por Jmeter. Elaborado por: los autores.

4.16 Análisis de resultados

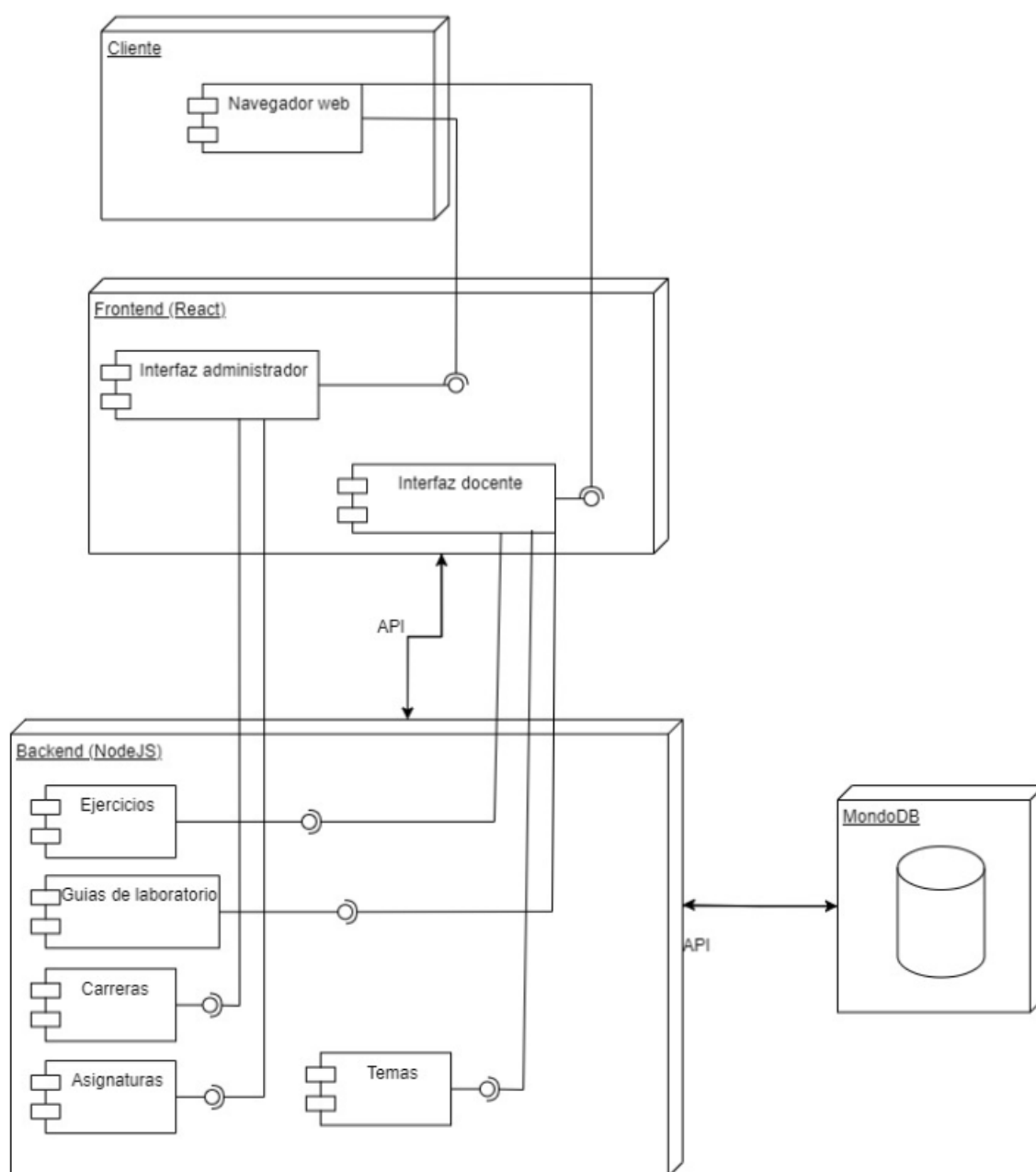
En base a las pruebas presentadas con anterioridad, se puede dar como efectivo el uso de aplicativo tanto en su funcionalidad, codificación y en la carga de usuarios optima siendo el caso más aproximado el de 25 usuarios concurrentes, donde no se espera un uso excesivo del aplicativo, debido a que su pico más alto de concurrencia sería al inicio de cada periodo académico. En todo momento el servidor mantiene una carga de contenido rápida y sin problemas de rendimiento.

4.17 Diagrama de implementación

El diagrama de implementación muestra la arquitectura de ejecución del aplicativo y la relación de los componentes con la arquitectura planteada. Ver figura 71.

Figura 71

Diagrama de implementación



Nota. Diagrama de implementación del aplicativo web. Fuente: los autores.

CAPITULO V

IMPLEMENTACIÓN

En este capítulo se presentan las configuraciones y pasos que se siguieron con el fin de implementar el aplicativo web en un servidor en internet.

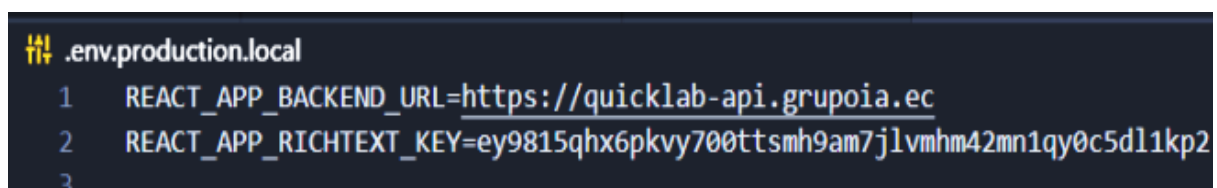
5.1. Build de la aplicación frontend

El primer paso para levantar el aplicativo en producción es mediante la ejecución del comando “npm build” en el frontend del aplicativo, con el fin de que todo el aplicativo sea comprimido en una carpeta llamada “build” el cual solo contendrá archivos HTML, css y js, los cuales no tendrán paquetes que fueron usados para el desarrollo. A fin de poder identificar la carpeta build del aplicativo se la renombro como “herramienta-colaborativa-ui”.

Cabe mencionar que se debe crear un archivo llamado “.env.production.local”, el cual contendrá dos variables una para el url de las API y otra para el api key que nos brinda la página de tinymce.

Figura 72

Variables de entorno frontend



```
1 REACT_APP_BACKEND_URL=https://quicklab-api.grupoia.ec
2 REACT_APP_RICHTEXT_KEY=ey9815qhx6pkvy700ttsmh9am7j1vmhm42mn1qy0c5d11kp2
3
```

Nota. Variables de entorno contenidas en env.production.local. Elaborado por: los autores.

5.2. Despliegue de backend

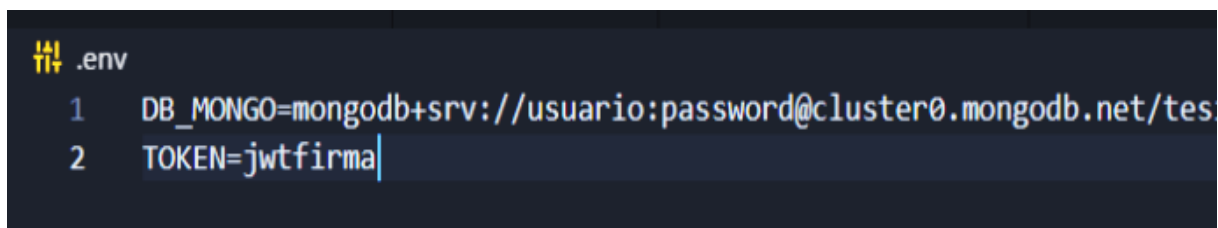
El despliegue del backend se hizo mediante el paquete “PM2”, el cual permite manejar los estados del aplicativo de una forma sencilla. El comando usado para levantar el API fue

“ pm2 start server/server.js --name "quicklab-api" ”, el cual levanta el aplicativo en el puerto configurado en este caso el 3001. Para verificar que el proceso está levantado se ejecutó el comando “pm2 list”.

En este punto también se requiere de un archivo que contenga las variables de entorno denominado “.env.”, dichas variables contendrán el url de conexión a la base de datos de MongoDB y la palabra secreta para los tokens JWT. No es necesario crear las tablas de la base de datos gracias al uso de Mongoose, cabe mencionar que el nombre de la base de datos a la que se hace referencia se encuentre en el url de conexión.

Figura 73

Variables de entorno backend



```
1 DB_MONGO=mongodb+srv://usuario:password@cluster0.mongodb.net/tes
2 TOKEN=jwtfirma|
```

Nota. Variables de entorno contenidas en env. Elaborado por: los autores.

5.3. Despliegue en internet

El claustro docente de programación nos proporcionó dos dominios los cuales apuntan a un servidor de Amazon web services, dichos dominio son:

- <https://quicklab.grupoia.ec/>: frontend
- <https://quicklab-api.grupoia.ec/>: backend

El despliegue del aplicativo en internet se hizo mediante el uso de “nginx” como servidor, para lo cual se utilizó dos archivos de configuración.

El archivo de configuración del frontend está diseñado para que cada vez que el usuario visite la página “https://quicklab.grupoia.ec/ ” se cargue la carpeta “herramienta-colaborativa-ui” en el navegador del usuario.

Figura 74

Código configuración frontend

```
server {  
    if ($host = quicklab.grupoia.ec) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
  
    listen 80;  
  
    server_name quicklab.grupoia.ec;  
    return 404; # managed by Certbot  
}
```

Nota. Código de configuración frontend para la implementación. Elaborado por: los autores.

Mientras el archivo de configuración del backend está diseñado para que cada vez que se haga una petición a “https://quicklab-api.grupoia.ec/”, se realiza un proxy al puerto 3001, donde está levantando el backend levantado con pm².

Figura 75

Código configuración backend

```
server {  
    if ($host = quicklab-api.grupoia.ec) {  
        return 301 https://$host$request_uri;  
    } # managed by Certbot  
  
    server_name quicklab-api.grupoia.ec;  
    listen 80;  
    return 404; # managed by Certbot  
}
```

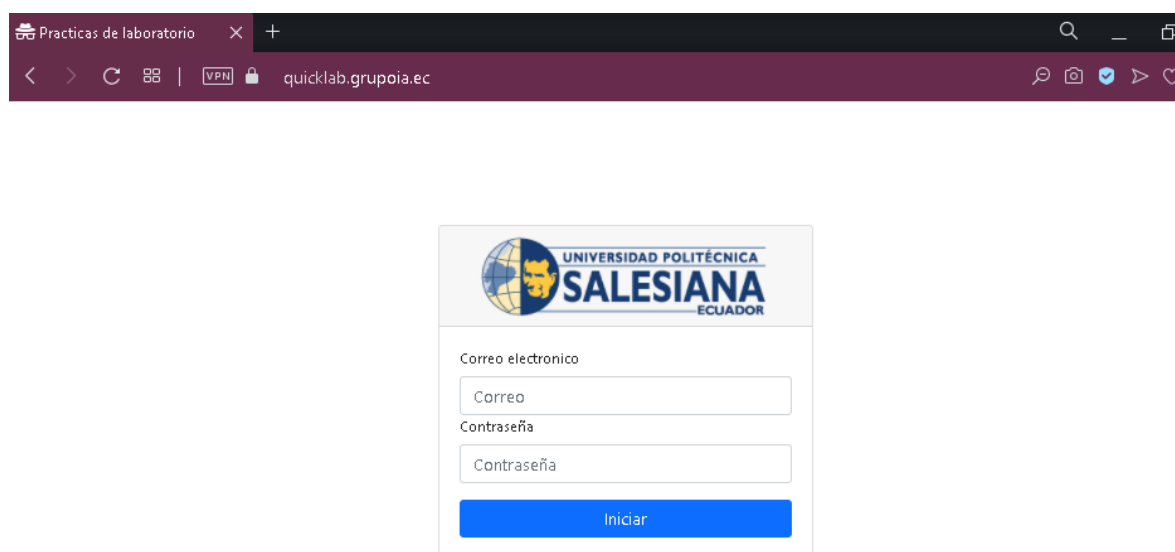
Nota. Código de configuración backend para la implementación. Elaborado por: los autores.

5.4 Aplicativo funcionando en internet

Una vez realizadas las configuraciones previamente descritas, se procede a comprobar el funcionamiento del aplicativo en el dominio público web “<https://quicklab.grupoia.ec>”, donde será usado por los docentes.

Figura 76

Página de inicio de sesión del aplicativo



Practicas de laboratorio

quicklab.grupoia.ec

UNIVERSIDAD POLITÉCNICA
SALESIANA
ECUADOR

Correo electronico

Correo

Contraseña

Contraseña

Iniciar

Nota. Página de inicio de sesión desde el dominio web donde se aloja la aplicación.

Elaborado por: los autores.

CONCLUSIONES

- Una etapa de análisis minuciosa es de gran importancia, la reunión con los involucrados del área donde será usado el aplicativo (claustro docente de programación), ayuda a entender el proceso actual, información usada y terminología, dicha información permitió enfocar el problema, a fin de proporcionar una solución adecuada.
- Los objetivos de construir e implementar el aplicativo se completaron de manera satisfactoria siguiendo las pautas establecidas en las etapas de análisis y diseño del presente proyecto, dando como resultado una aplicación web, la cual agiliza el proceso de creación de guías de laboratorio, manteniendo una base de ejercicios creados por los docentes, los cuales serán usados en cada periodo académico, el aplicativo está implementado en un dominio web para su acceso por parte de cualquier docente de la Universidad Politécnica Salesiana.
- Las pruebas realizadas sobre el aplicativo constataron que el mismo, posee un tiempo de respuesta promedio de 2000 milisegundos para las peticiones con una concurrencia de 25 usuarios al mismo tiempo, gracias al uso de NodeJS y React, adicionalmente el amplio catálogo de librerías de React, garantiza que funciones repetitivas como el inicio de sesión estén probados y libres de errores, garantizando un correcto funcionamiento.
- El uso de herramientas que ayuden en la gestión y seguimiento del trabajo son fundamentales cuando se realiza un trabajo colaborativo y cambiante mediante la metodología SCRUM, debido a ello el uso de GitHub y Jira para mantener un repositorio de código y dar seguimiento de las iteraciones respectivamente, es de suma importancia, en especial GitHub el cual no funge como un simple repositorio de código fuente sino que ayuda a mantener el avance de los cambios para futuras revisiones por parte de otros desarrolladores.

- El concepto de herramienta colaborativa usado en el desarrollo del presente proyecto ayuda a los docentes no solo a agilizar el proceso de creación de guías de laboratorio, sino ofrecer un ambiente de colaboración donde los usuarios pueden crear sus ejercicios y acceder a una colección de ejercicios creados en periodos posteriores, generando retroalimentación mediante un sistema de calificación a fin de mejorar la calidad de las guías de laboratorio en cada periodo académico.
- Un aplicativo minimalista y con opciones focalizadas fue una parte central del desarrollo, proporcionando al usuario cuadros de edición de texto fáciles y con todas las opciones de editores de texto en el mercado, con la finalidad de que la creación de ejercicios en la aplicación sea lo más familiar y fácil posible.

RECOMENDACIONES

- Es recomendable realizar un estudio previo en la selección de las herramientas de desarrollo, ya que dependiendo del objetivo del aplicativo es necesario la adecuada selección de lenguajes de programación, bases, marcos de trabajo. Ponderando las ventajas y desventajas que ofrecen, junto a la experiencia acumulada y las tendencias del mercado, seleccionar los más adecuados para cumplir con los objetivos del aplicativo.
- El aplicativo en el presente proyecto está enfocado para el claustro docente de programación, pero su uso es recomendable para cualquier claustro docente perteneciente a la Universidad Politécnica Salesiana, independientemente de la carrera a la cual pertenezca.
- Se recomienda agregar una funcionalidad la cual permita exportar los ejercicios creados en el aplicativo a un formato que pueda ser cargado en la plataforma AVAC institucional de la Universidad Politécnica Salesiana.
- El presente proyecto posee la funcionalidad de crear citas bibliográficas mediante un formulario limitado solo a libros y páginas web asociado a la estructura APA 7, por ende, es recomendable la integración con una aplicación externa que gestione las referencias bibliográficas como Mendeley, con el fin de importarlas al aplicativo y no tener que crearlas.

REFERENCIAS

Artículos Académicos

- Agus, I., Destiawat, F., & Dhika, H. (2019). Perbandingan Cloud Computing Microsoft Onedrive, Dropbox, Google Drive. *Faktor Exacta*, 12(1), 20-27.
<https://doi.org/10.30998/faktorexacta.v12i1.3631>.
- Castillo, J. N., Garcés, J. R., Navas, M. P., Segovia, D. F. J., & Naranjo, J. E. A. (2017). Base de Datos NoSQL: MongoDB vs. Cassandra en operaciones CRUD (Create, Read, Update, Delete). *Revista Publicando*, 4(11 (1)), 79-107.
- Challenger, I., Diaz, Y. y Becerra, R. (2014). El lenguaje de programación Python.
- Coyle, J. (2007), "Wikis in the college classroom: a comparative study of online and face to face group collaboration at a private liberal arts university", unpublished doctoral dissertation, Kent State University College and Graduate School of Education, Health and Human Services, Kent, OH.
- Duarte, D. L., and Tennant-Snyder, N., *Mastering virtual teams: Strategies, tools, and techniques that succeed*, San Francisco, CA: Jossey Bass, 2000, pp.40.
- García Sandoval, M. G., Ariza Torrado, H., Pinzón, M. L. y Flórez Fuentes, A. S. (2015). Buenas prácticas aplicadas a la implementación colaborativa de aplicativos Web. *Revista Mundo FESC*, 10, pp. 27-30.
- Kai-Wai Chu, S., & Kennedy, D. M. (2011). Using online collaborative tools for groups to co-construct knowledge. *Online Information Review*, 35(4), 581-597.
<https://doi.org/10.1108/14684521111161945>
- Lopez-Pellicer, F. J., Béjar, R., Latre, M. A., Nogueras-Iso, J., & Zarazaga-Soria, F. J. (2015, July). GitHub como herramienta docente. In *Actas de las XXI Jornadas de la Enseñanza Universitaria de la Informática* (pp. 66-73). Universitat Oberta La Salle.

- Mariño, S. I., & Alfonzo, P. L. (2014). Implementación de SCRUM en el diseño del proyecto del Trabajo Final de Aplicación. *Scientia et technica*, 19(4), 413- 418.
- Massey, A. P., Chapter 17, Collaborative Technologies, Handbook on Decision Support Systems 1, Editors: F. Burstein, C.W. Holsapple, Springer, 2008, pp. 345-351.
- Otero, J. J. E., Sanz, E. P., & Otero, C. A. E. (2007). Uso de herramientas colaborativas que reducen la carga de gestión en la docencia. *Actas de las XIII Jornadas de Enseñanza Universitaria de Informática*, Jenui, 301-308.
- Rivas, C. I., Corona, V. P., Gutiérrez, J. F., & Hernández, L. (2015). Metodologías actuales de desarrollo de software. *Revista de Tecnología e Innovación*, 2(5), 980-986.
- Rosado-Gómez, A., Quintero-Duarte, A., & Meneses-Guevara, C. D. (2012). Desarrollo ágil de software aplicando programación extrema. *Revista Ingenio*, 5(2), 24-29.
- Salazar, J. C., Casallas, Á. T., Linares, J. C., Lozano, A., & Valbuena, Y. L. (2018). SCRUM versus XP: similitudes y diferencias. *Tecnología Investigación Y Academia*, 6(2), 29-37.
- Trigás Gallego, M. (2012). Metodología scrum. Pardo, M. R. V., Tapia, J. A. H., Moreno, A. S. G., & Sánchez, L. F. V. (2018). Comparación de tendencias tecnológicas B. en aplicaciones web.
- Sucre, F. S., & Chirinos, D. P. (2014). Marco de trabajo para gestionar las competencias laborales. *Enlace: Revista Venezolana de Información, Tecnología y Conocimiento*, 11(3), 11-32.

Libros

- Alonso, F., & others. (2005). *Introducción a la ingeniería del software*. Delta Publicaciones.
- Eslava, V. (2013). *El nuevo PHP. Conceptos avanzados*: Bubok Publising S. L.
- Flanagan, D. (2007). *JavaScript. La Guía Definitiva*: Anaya.

González Duque, R. (2014). Python para todos.

Maza, M. Á. S. (2012). JavaScript. Innovación y Cualificación..

Mohedano, J., Saiz, J. M. y Salazar, P. (2012). Iniciación a JavaScript. Ministerio de Educación.

Pressman, R. S., & Troya, J. M. (1988). Ingeniería del software.

Ramos, D., Noriega, R., Laínez, J. R., & Durango, A. (2017). Curso de Ingeniería de Software:
2aEdición. IT Campus Academy.

Sommerville, I. (2005). Ingeniería del software. Pearson educación.

Páginas web

Claustro Docente - UPS. (2017). Claustro docente. Extraído el 24 de Marzo, 2021, de
ups.edu.ec. Sitio web: <https://www.ups.edu.ec/destacado?entryId=8369909>.

MongoDB. (2021). Compass. Extraído el 18 de Mayo, 2021, de www.mongodb.com.

Sitio web: <https://www.mongodb.com/products/compass>

MongoDB. (2021). La base de datos líder del mercado para aplicaciones modernas. Ex traído
el 18 de Mayo, 2021, de www.mongodb.com. Sitio web:

<https://www.mongodb.com>.

PostMan. (2021). API-First Development. Extraído el 18 de Mayo, 2021, de

www.postman.com. Sitio web: <https://www.postman.com/use-cases/api-first-development/>

Schwaber, K., Sutherland, J. (2020). La Guía de SCRUM. Extraído el 24 de Marzo, 2021, de
SCRUMguides.org. Sitio web:

<https://SCRUMguides.org/docs/SCRUMguide/v2020/2020-SCRUM-Guide-Spanish-Latin-South-American.pdf>

Visual Studio Code. (2021). Get Started with Visual Studio Code. Extraído el 18 de Mayo, 2021, de Code.visualstudio.com. Sitio web: <https://code.visualstudio.com/learn>.

Wells, Don. "Extreme Programming: A gentle introduction." Internet URL. www.Extremeprogramming.Org (2001).

Tesis

Román Seneque, N. (2014). Administración en MongoDB.

Saks, E. (2019). JavaScript Frameworks: Angular vs React vs Vue.

Rodríguez Flores, G. (2017). Desarrollo de una aplicación web con Node.js para la monitorización en tiempo real de un electrocardiograma.

ANEXOS

Viabilidad económica

Detalles de costos de hardware

Recursos de hardware			
Recurso	Cantidad	Valor unitario	Valor Total
Ordenador portátil HP	1	\$1.200,00	\$1.200,00
Ordenador de sobremesa	1	\$500,00	\$500,00
Total			\$160,00

Detalles de costos de software

Recursos de software				
Recurso	Descripción	Cantidad	Valor unitario	Valor total
Licencia Windows 10	Windows 10 Pro	2	\$80,00	\$160,00
Visual Studio Code	Licencia MIT	2	\$0,0	\$0,0
GitHub	OpenSource	2	\$0,0	\$0,0
MongoDB Atlas	Capa gratuita	1	\$0,0	\$0,0
Total				\$160,00

Detalles de costos de comunicación

Recursos de comunicación			
Definición	Cantidad	Valor mensual	Valor semestral
Conexión a Internet	2	\$30,00	\$180,00
Total			\$360,00

Costos recursos humanos

Recursos humanos				
Recurso	Descripción	Cantidad	Valor mensual	Valor semestral (total)
Programador	Salario mensual de programador	2	\$1600,00	\$9600,00
Servicios básicos	Electricidad, Agua	2	\$420,00	\$420,00
Gastos operativos	Alimentación, movilización, impresiones, etc.	2	\$100,00	\$600,00
Total				\$10620,00

Detalles de costos extras relacionados

Recursos extra relacionados			
Definición	Cantidad	Valor unitario	Valor total
Matrícula e ingreso a Unidad de Titulación	2	\$550,00	\$968,00
Total			\$968,00

Características de usuarios

Usuario del aplicativo

Usuario	Administrador.
Formación	Persona no técnica
Habilidades	Conocimiento del proceso de asignación de materias, administración de usuarios.
Actividades	Control y manejo del sistema en general

Usuario del aplicativo

Usuario	Docente.
Formación	Persona no técnica.
Habilidades	Manejo de computadores y aplicaciones de edición de texto.
Actividades	Creación de ejercicios y soluciones para las guías de laboratorio, creación de citas bibliográficas.

Calendario Sprints

Calendario de sprints del mes de diciembre

Diciembre				
Lunes	Martes	Miércoles	Jueves	Viernes
	1	2	3	4
7	8	9	10	11
14	15	16	17	18
21 Sprint Planning (Sprint 1)	22 Reunión diaria	23 Reunión diaria	24 Reunión diaria	25 Sprint Review (Sprint 1) Sprint Retrospective (Sprint 1)
28 Sprint Planning (Sprint 2)	29 Reunión diaria	30 Sprint Review (Sprint 2) Sprint Retrospective (Sprint 2)		

Calendario de sprints del mes de enero

Enero				
Lunes	Martes	Miércoles	Jueves	Viernes

			31 Sprint Planning (Sprint 3)	1
4 Reunión diaria	5 Reunión diaria	6 Reunión diaria	7	8 Sprint Review (Sprint 3) Sprint Retrospective (Sprint 3)
11	12	13	14	15 Sprint Planning (Sprint 4)
18 Reunión diaria	19 Reunión diaria	20 Reunión diaria	21 Reunión diaria	22 Sprint Review (Sprint 4) Sprint Retrospective (Sprint 4)
25	26	27	28	29 Sprint Planning (Sprint 5)

Calendario de sprints del mes de febrero

Febrero				
Lunes	Martes	Miércoles	Jueves	Viernes
1 Reunión diaria	2 Reunión diaria	3 Reunión diaria	4 Reunión diaria	5 Sprint Review (Sprint 5) Sprint Retrospective (Sprint 5)

8	9	10	11	12 Sprint Planning (Sprint 6)
15 Reunión diaria	16 Reunión diaria	17 Reunión diaria	18 Reunión diaria	19 Sprint Review (Sprint 6) Sprint Retrospective (Sprint 6)
22	23	24	25	26 Sprint Planning (Sprint 7)

Calendario de sprints del mes de marzo

Marzo				
Lunes	Martes	Miércoles	Jueves	Viernes
1 Reunión diaria	2 Reunión diaria	3 Reunión diaria	4 Reunión diaria	5 Sprint Review (Sprint 7) Sprint Retrospective (Sprint 7)
8	9	10	11	12 Sprint Planning (Sprint 8)
15 Reunión diaria	16 Reunión diaria	17 Reunión diaria	18 Reunión diaria	19 Sprint Review (Sprint 8) Sprint Retrospective (Sprint 8)
22 Sprint Planning (Sprint 9)	23 Reunión diaria	24 Reunión diaria	25 Reunión diaria	26 Reunión diaria

29 Sprint Review (Sprint 9) Sprint Retrospective (Sprint 9)	30	31		
--	-----------	-----------	--	--

Calendario de sprints del mes de abril

Abril				
Lunes	Martes	Miércoles	Jueves	Viernes
			1	2 Sprint Planning (Sprint 10)
5 Reunión diaria	6 Reunión diaria	7 Reunión diaria	8 Reunión diaria	9 Sprint Review (Sprint 10) Sprint Retrospective (Sprint 10)
12	13	14	15	16 Sprint Planning (Sprint 11)
19 Reunión diaria	20 Reunión diaria	21 Reunión diaria	22 Reunión diaria	23 Sprint Review (Sprint 11) Sprint Retrospective (Sprint 11)
26	27	28	29	30